



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Escuela Académico Profesional de Ingeniería de Sistemas

**Un sistema de generación de horarios para la
enseñanza de pregrado en universidades peruanas
mediante algoritmos genéticos**

TESIS

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Sherly Patricia BLAZ ARISTO

ASESOR

David Santos MAURICIO SÁNCHEZ

Lima, Perú

2016



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Blaz, S. (2016). *Un sistema de generación de horarios para la enseñanza de pregrado en universidades peruanas mediante algoritmos genéticos*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Académico Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
Escuela Académico Profesional de Ingeniería de Sistemas

Acta de Sustentación de Tesis

Siendo las 19.15 horas del día 27 de Mayo del año 2016, se reunieron los docentes designados como miembros de Jurado de la Tesis, presidido por el Lic. José Cesar Piedra Isusqui (Presidente), el Lic. Jorge Luis Chávez Soto (Miembro) y el Dr. David Santos Mauricio Sánchez (Miembro Asesor) para la sustentación de la Tesis Intitulada: **"UN SISTEMA DE GENERACIÓN DE HORARIOS PARA LA ENSEÑANZA DE PREGRADO EN UNIVERSIDADES PERUANAS MEDIANTE ALGORITMOS GENÉTICOS"**. Por la Bachiller: BLAZ ARISTO; SHERLY PATRICIA; para optar el Título Profesional de Ingeniero de Sistemas.

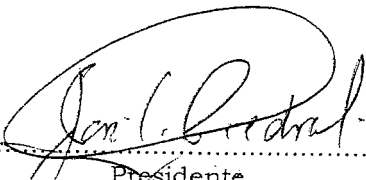
Acto seguido de la exposición de la Tesis, el presidente invitó al graduado a dar las respuestas a las preguntas establecidas por los Miembros del Jurado.

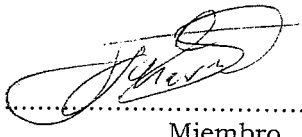
EL graduado en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

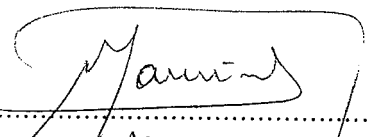
Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el graduado obtuvo la nota de 19 (En letras) diecinueve.

A continuación el Presidente de Jurados el Lic. José Cesar Piedra Isusqui, declara al graduado **Ingeniero de Sistemas**.

Siendo las 20.15 Horas, se levantó la sesión.


.....
Presidente
Lic. José Cesar Piedra Isusqui


.....
Miembro
Lic. Jorge Luis Chávez Soto


.....
Asesor
Dr. David Santos Mauricio Sánchez

© Sherly Blaz y Kenny Francia, 2016.

Todos los derechos reservados.

Este trabajo está dedicado a nuestras familias que siempre han mantenido la esperanza en nosotros y han confiado plenamente en nuestra capacidad de superación.

AGRADECIMIENTOS

Al profesor Dr. David Mauricio, por su orientación y dedicación para que este trabajo cumpla con los objetivos trazados.

A nuestras familias por apoyarnos a lo largo de la carrera y, más aún, a lo largo del trabajo que representa la culminación de la misma.

A todas aquellas personas que indirectamente nos ayudaron para culminar este trabajo y que muchas veces constituyen un invalorable apoyo.

Y por encima de todo doy gracias a Dios.

Un Sistema de Generación de Horarios para la Enseñanza de Pregrado en Universidades Peruanas mediante Algoritmos Genéticos

RESUMEN

Este proyecto intenta dar solución al problema de generación de horarios para los cursos universitarios de pregrado en las Universidades Peruanas, en la cual cada encargado de la elaboración de horarios debe realizar una laboriosa tarea de asignación de salones y docentes en un periodo específico para los dictados de clases de las asignaturas que se imparten en un ciclo académico. La presente investigación es un esfuerzo para facilitar dicha labor manual, ya que el procedimiento actual en muchas Universidades del Perú requiere muchos recursos y un gran esfuerzo de parte de los responsables, además de generar horarios no factibles y no aprovechar de manera óptima los recursos de la institución. En este trabajo se propone un sistema inteligente de generación de horarios basado en algoritmos genéticos, el cual fue adaptado para poder cumplir con los requisitos específicos impuestos por cada Facultad de las diferentes Universidades del Perú y así satisfacer a los usuarios involucrados. En nuestras pruebas se consideró como caso de estudio la Facultad de Ingeniería de Sistemas de la Universidad Nacional Mayor de San Marcos, obteniendo como resultado un horario con cero violaciones de restricciones obligatorias y reduciendo en lo más posible las violaciones de las restricciones blandas y así superando la generación de horarios elaborado de forma manual que actualmente se lleva a cabo.

Palabras Claves: Problema de horarios de cursos universitarios (UCTP), algoritmos genéticos, algoritmos de búsqueda local.

A Timetable Generation System for Teaching Undergraduate Peruvian universities by Genetic Algorithms

ABSTRACT

This project tries to solve the problem of generating schedules for university undergraduate courses in Peruvian universities, in which each responsible for drawing up schedules to perform a laborious task allocation of classrooms and teachers in a specific period for the dictates class of subjects taught in an academic year. This research is an effort to facilitate such manual labor, since the current procedure in many Universities of Peru requires many resources and a great effort on the part of those responsible, and generate infeasible schedule and do not make best use of resources of the institution. This thesis presents an intelligent system for generating schedules based on genetic algorithms, which was adapted to meet the specific requirements of each faculty of various universities of Peru and satisfy the users involved. In our tests it was considered as a case study the Faculty of Engineering Systems at the National University of San Marcos, resulting in a schedule with zero violations of mandatory restrictions and reducing as much as possible violations of the soft constraints and surpassing generating schedules manually developed which is currently performed.

Key words: University course timetabling problem (UCTP), genetic algorithms, local search algorithms.

ÍNDICE

Lista de Figuras	xi
Lista de Tablas	xiv
CAPÍTULO 1: INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Definición del Problema	3
1.2.1 Restricciones Obligatorias	3
1.2.2 Restricciones Deseables	4
1.3 Objetivos	5
1.3.1 Objetivo Principal	5
1.3.2 Objetivos Secundarios	5
1.4 Justificación	5
1.5 Alcances	8
1.6 Propuesta	8
1.7 Organización de la Tesis	9
CAPÍTULO 2: REVISIÓN DE ALGORITMOS PARA EL UCTP	10
2.1 Algoritmos para generación de horarios a lo largo de la historia	10
2.2 Algoritmos para Generación de Horarios	12
2.2.1 Un algoritmo genético extendido con búsqueda local y búsqueda guiada (EGSGA)	12
2.2.2 EL algoritmo Multiswap	15
2.2.3 Uso de las estructuras vecinales múltiples dentro de un enfoque memético basado en Tabú a problemas de horarios universitarios	17
2.2.4 Un nuevo algoritmo de búsqueda armónica	19
2.2.5 Un enfoque meta-heurístico híbrido	21
2.3 Benchmarking de algoritmos propuestos en la literatura	23
CAPÍTULO 3: ALGORITMOS GENÉTICOS CON BÚSQUEDA GUIADA Y BÚSQUEDA LOCAL DE SHENGXIANG YANG Y SADAF NASEEM JAT	25
3.1 Descripción del Problema	25
3.2 Descripción del Algoritmo	26
3.3 Operador de Cruzamiento	28
3.4 Búsqueda Local LS1	28

3.5 Extensión del GSGA	29
CAPÍTULO 4: ADAPTACIÓN DEL ALGORITMO GENÉTICO	31
4.1 Problema.....	31
4.2 Modelo General	32
4.2.1 Parámetros	32
4.2.2 Variables	36
4.2.3 Restricciones.....	36
4.3 Modelo Particular	39
4.3.1 Parámetros	39
4.3.2 Variables	40
4.4 Función Objetivo	40
4.5 Representación del Problema	41
4.6 Algoritmo Adaptado	42
4.6.1 Descripción del Algoritmo	44
CAPÍTULO 5: SISTEMA DE GENERACIÓN DE HORARIOS	55
5.1 Definición de la Metodología de a usar.....	55
5.2 Requerimientos	56
5.3 Análisis y Diseño.....	72
5.4 Arquitectura.....	72
CAPÍTULO 6: ESTUDIO EXPERIMENTAL	73
6.1 Organización de la Facultad	73
6.1.1 Carreras, Años y Ciclos	73
6.1.2 Asignaturas	73
6.1.3 Grupos	74
6.1.4 Clases.....	74
6.1.5 Turnos.....	75
6.1.6 Salones.....	75
6.1.7 Características.....	76
6.1.8 Horarios de dictado de Clases	76
6.2 Análisis de Sensibilidad de los Parámetros	77
6.2.1 Prueba de Número de Generaciones.....	77
6.2.2 Prueba de Porcentajes	77
6.3 Caso de Pruebas.....	78

6.3.1 Descripción de la Instancia.....	79
6.3.2 Criterios de Comparación.....	79
6.4 Análisis de Resultados.....	80
CAPÍTULO 7: CONCLUSIONES Y TRABAJOS FUTUROS	83
7.1 Conclusiones.....	83
7.2 Trabajos Futuros	84
REFERENCIAS BIBLIOGRÁFICA	85

Lista de figuras

Figura 1.1. Número de Universidades, por año censal, según tipo de Universidad	1
Figura 1.2. Población Universitaria por año censal y tasa de crecimiento anual según tipo de Universidad	2
Figura 1.3. Total de docentes universitarios, por condición laboral y categoría según tipo de Universidad	2
Figura 1.4. Número de Facultades y Carreras Profesionales que ofrecen las Universidades Peruanas	6
Figura 1.5. Número de Universidades creadas según tipo de universidad 1551-2011	6
Figura 1.6. Porcentaje de Docentes Universitarios, por condición laboral según tipo de Universidad	7
Figura 1.7. Patrón de Arquitectura del Sistema Propuesto	8
Figura 2.1. Línea de tiempo de Algoritmos propuestos para solucionar el UCTP	12
Figura 2.2. Algoritmo Genético con Búsqueda Guiada (GSGA)	13
Figura 2.3. Algoritmo de Búsqueda Local (LS2)	14
Figura 2.4. Esquema del algoritmo propuesto	15
Figura 2.5. El pseudo-código del algoritmo MultiSwap	16
Figura 2.6. Algoritmo Memético basado en Tabú	18
Figura 2.7. Algoritmo de búsqueda armónica	19
Figura 2.8. Pseudocódigo Genérico del algoritmo	22
Figura 2.9. Pseudocódigo del algoritmo meta-heurístico híbrido	22
Figura 3.1. Algoritmo Genético con Búsqueda Guiada y Búsqueda Local	27
Figura 3.2. Pseudocódigo del operador de Cruzamiento	28
Figura 3.3. Algoritmo de Búsqueda Local LS1	29
Figura 3.4. Algoritmo de Búsqueda Local LS2	30
Figura 4.1. Periodos de Clase	39
Figura 4.2. Representación del gen	42
Figura 4.3. Algoritmo Genético Adaptado	43
Figura 4.4. Algoritmo de Inicialización de la población.	45
Figura 4.5. Cruzamiento	48

Figura 4.6 Algoritmo de Compactación	50
Figura 4.7 Algoritmo de mutación	51
Figura 4.8 Algoritmo LS1	52
Figura 4.9. Algoritmo LS2	53
Figura 5.1. 1 Los cinco flujos de trabajo sobre las cuatro fases: inicio, elaboración, construcción y prueba	55
Figura 5.2. Diagrama de paquetes	56
Figura 5.3. Diagrama de perfiles de usuarios	58
Figura 5.4. Diagrama de Casos de Uso del Módulo Configuración de Horarios	59
Figura 5.5. Gestión de Ciclo	59
Figura 5.6. Mantenimiento de Dictados de Clase	60
Figura 5.7. Configuración de Periodo de Dictado por Ciclo	60
Figura 5.8. Configuración de Parámetros del algoritmo	61
Figura 5.9. Disponibilidad de Docente	61
Figura 5.10. Clases de Preferencia de un Docente	62
Figura 5.11. Asignar Dictado de Clase	62
Figura 5.12. Diagrama de Casos de Uso del Módulo Seguridad	63
Figura 5.13. Loguear Usuario	63
Figura 5.14. Cambiar contraseña	64
Figura 5.15. Cambiar de Perfil	64
Figura 5.16. Gestionar Opciones por Perfil	65
Figura 5.17. Gestionar Perfiles por Usuario	65
Figura 5.18. Diagrama de Casos de Uso del Módulo de Mantenimiento	66
Figura 5.19. Mantenimiento de Dirección de Escuela	67
Figura 5.20. Mantenimiento de Plan de Estudios	67
Figura 5.21. Mantenimiento de Asignaturas	67
Figura 5.22. Mantenimiento de Clases	68
Figura 5.23. Mantenimiento de Docentes	69
Figura 5.24. Mantenimiento de Salones	69
Figura 5.25. Mantenimiento de Laboratorios	70
Figura 5.26. Diagrama de Casos de Uso del Módulo de Elaboración de Horarios	70
Figura 5.27. Gestión de Horarios	71
Figura 5.28. Consultar Horario	71

Figura 6.1. Diagrama de Procesos de la Fisi	76
Figura 6.2. Periodos de Clase	79
Figura 6.3. Comparación del número de violaciones duras y blandas entre el horario manual y el algoritmo propuesto	82

Lista de tablas

Tabla 2.1 Comparación del número de violaciones de restricciones blandas del algoritmo EGSGA con otros algoritmos de la literatura	15
Tabla 2.2 Comparación del número de violaciones de restricciones blandas Multiswap con otros algoritmos de la literatura	17
Tabla 2.3 Comparación del número de violaciones de restricciones blandas del algoritmo HS con otros algoritmos de la literatura	21
Tabla 2.4 Comparación del número de violaciones de restricciones blandas del algoritmo meta heurístico híbrido con otros algoritmos de la literatura	23
Tabla 2.5 Características del dataset de Socha	23
Tabla 2.6 Comparación del número de violaciones de restricciones blandas de los algoritmos presentados	24
Tabla 4.1 Matriz Dictado de Clase - Conflictos	34
Tabla 4.2. Matriz Docente - Periodo	34
Tabla 4.3. Matriz Dictado de Clase – Salón	35
Tabla 4.4. Matriz Clase - Periodo	35
Tabla 4.5. Matriz Salón - Periodo	36
Tabla 4.6. Matriz de Representación del Problema	41
Tabla 5.1. Lista de Perfiles	57
Tabla 6.1 Valores de los parámetros del algoritmo	78
Tabla 6.2. Las 12 mejores combinaciones de parámetros	78
Tabla 6.3 Medición de indicadores para los horarios 2015-I	80

Capítulo 1: Introducción

1.1 Antecedentes

Las Universidades del Perú han ido aumentando considerablemente durante la última década, según el último Censo Nacional Universitario realizado en el 2010 por la INEI [41] hay 782 970 alumnos de pre grado en 100 universidades que vienen funcionando formalmente en el Sistema Universitario Peruano, ver Figura 1.1.

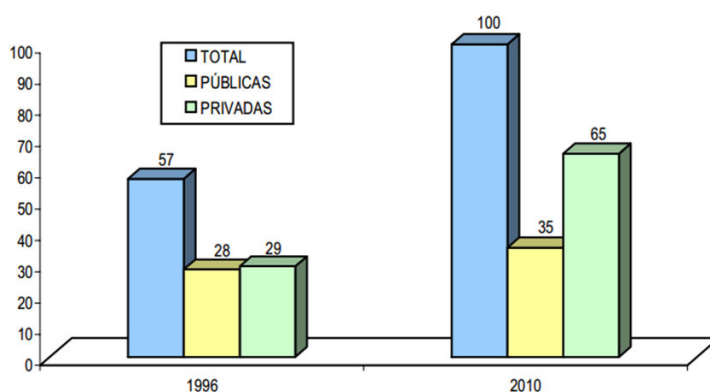


Figura 1.1 Número de Universidades, por año censal, según tipo de Universidad [41].

La población involucrada directamente en el complejo universitario peruano, constituida por estudiantes (de pre y post grado), por docentes y personal administrativo y de servicios, se aproxima al millón de personas. En un lapso de 14 años (1996 al 2010) se ha incrementado en 2,4, pasando de 389 316 a 937 430 personas. Este crecimiento es diferencial en la gestión pública o privada. Los estudiantes de pre grado son quienes representan el mayor volumen poblacional, ascienden a 782 970 personas en el año 2010, los que constituyen el 83,5% de la población universitaria. En el año 1996 los estudiantes de las universidades públicas conformaban la mayor parte (59,6% del total), la universidad privada, en virtud de su acelerado crecimiento logra posicionarse en el 2010 como la de mayor población estudiantil (60,5%). En el periodo en referencia los estudiantes de post grado se multiplicaron por 5,2 veces, pasando de 10 818 en 1996 a 56 358 en el 2010. Entre el año 1996 y el 2010, los docentes universitarios aumentaron de 25 795 a 59 085 personas, los trabajadores administrativos

pasaron en el mismo periodo de 16 989 a 39 017, las tasas de crecimiento para ambos estamentos fueron de 5,2% y 6,1% anual, ver Figura 1.2.

TIPO DE UNIVERSIDAD	NÚMERO DE UNIVERSIDAD	ALUMNOS		DOCENTES UNIVERSITARIOS	PERSONAL ADMINISTRATIVO Y DE SERVICIOS
		PRE GRADO	POST GRADO		
AÑO 2010					
TOTAL	100	782 970	56 358	59 085	39 017
PÚBLICAS	35	309 175	24 591	21 434	19 961
PRIVADAS	65	473 795	31 767	37 651	19 056
AÑO 1996					
TOTAL	57	335 714	10 818	25 795	16 989
PÚBLICAS	28	199 943	7 109	16 096	11 708
PRIVADA	29	135 771	3 709	9 699	5 281
TASA DE CRECIMIENTO ANUAL PERIODO 1996-2010					
TOTAL	4,2	6,2	12,4	5,2	6,1
PÚBLICAS	1,6	3,1	9,2	1,4	3,9
PRIVADA	6,0	9,3	16,5	9,1	9,6

Figura 1.2 Población Universitaria por año censal y tasa de crecimiento anual según tipo de Universidad [41].

Las Facultades de las Universidades cuentan con una planilla de docentes de acuerdo a la condición laboral y a la categoría a la que pertenecen, es por ello que los docentes pueden dictar clases en diferentes universidades en las que se encuentren integrados. Según el INEI [41], hay 59 085 docentes universitarios, 36% en Universidades Públicas y el 64% en Universidades Privadas, ver Figura 1.3.

TIPO DE UNIVERSIDAD	CONDICIÓN LABORAL																
	TOTAL	CATEGORÍA DEL DOCENTE ORDINARIO					CLASIFICACIÓN DE DOCENTE EXTRAORDINARIO					CATEGORÍA DEL DOCENTE CONTRATADO					JEFE DE PRÁCTICA
		TOTAL	PRIN- CIPAL	ASO- CIADO	AUXI- LIAR	NINGUNO	TOTAL	EMERITO	HONO- RIARIO	INVESTI- GADOR	VISI- TANTE	TOTAL	PRIN- CIPAL	ASO- CIADO	AUXI- LIAR	NINGUNO	
TOTAL	59 085	20 486	7 034	7 662	5 424	366	368	43	65	86	174	34 422	4 480	2 885	12 811	14 246	3 809
	100,0	34,7	34,3	37,4	26,5	1,8	0,6	11,7	17,7	23,4	47,3	58,3	13	8,4	37,2	41,4	6,4
PÚBLICAS	21 434	15 726	5 472	6 176	4 008	70	68	6	14	16	32	4 417	157	242	3 353	665	1 223
	100,0	73,4	34,8	39,3	25,5	0,4	0,3	8,8	20,6	23,5	47,1	20,6	3,6	5,5	75,9	15,1	5,7
PRIVADAS	37 651	4 760	1 562	1 486	1 416	296	300	37	51	70	142	30 005	4 323	2 643	9 458	13 581	2 586
	100,0	12,6	32,8	31,2	29,7	6,2	0,8	12,3	17	23,3	47,3	79,7	14,4	8,8	31,5	45,3	6,9

Figura 1.3 Total de docentes universitarios, por condición laboral y categoría según tipo de Universidad [41].

Para cada periodo lectivo, se debe elaborar la carga horaria docente considerando su condición laboral (ordinario, extra ordinario, contratado o jefe de práctica), su categoría (principal, asociado o auxiliar) y su modalidad (tiempo completo o parcial).

Los docentes brindan un rango de horas de disponibilidad para la elaboración de horarios, cuya cantidad de horas mínimas que se solicite dependerá de las normativas

que tenga la Facultad, además la carga horaria se debe formalizar mediante una Resolución de la Facultad.

Desde siempre las Universidades del Perú presentan dificultades para la elaboración de horarios de clases debido a que esta actividad principalmente se realiza de forma manual y considera diferentes recursos (fijos y variables) como por ejemplo: docentes, alumnos, asignaturas, salones, etc.

Las Universidades del Perú, en su mayoría, no cuentan con un sistema que los apoye a la elaboración de horarios además que la tecnología promedio con la que se cuenta es una PC de doble núcleo para requerimientos de una oficina ordinaria, es por eso que se apoyan con documentos en Excel y con documentos físicos de información de los recursos que se requieren para la elaboración de horarios.

1.2 Definición del Problema

La programación de horarios de cursos universitarios es un problema complejo, especialmente por tener un gran número de restricciones las cuales son definidas de forma particular por cada institución y por requerir un alto esfuerzo computacional. El problema consiste en dado m : número de dictados de clases $C = \{c_1, c_2, \dots, c_m\}$, z : número de salones $S = \{s_1, s_2, \dots, s_z\}$, p : número de docentes $D = \{d_1, \dots, d_p\}$, y : número de periodos $T = \{t_1, \dots, t_y\}$ realizar la programación de actividades académicas, asignando de manera óptima los recursos, como por ejemplo programar un dictado de clase y asignar a este su respectivo docente, salón y periodo. Estos recursos se asignarán cumpliendo restricciones obligatorias o fuertes y se tratará de cumplir restricciones secundarias o débiles del problema, para agregar un alto grado de optimización y adaptabilidad a la solución. Sin embargo es suficiente cumplir las restricciones más fuertes para aceptar una solución como válida. Las restricciones tomadas en cuenta son las siguientes:

1.2.1 Restricciones Obligatorias

RO1. Una clase debe tener asignada, a lo más, un docente en un salón en un periodo específico.

RO2. Un docente debe tener asignado, a lo más, una clase en un salón en un periodo específico.

RO3. Un salón puede tener, a lo más, una clase asignada y un docente en un periodo específico.

RO4. Un docente tendrá asignado una lista de clases el cual dictará en el ciclo.

RO5. No se puede asignar una clase a un docente fuera de su disponibilidad horaria.

RO6. La capacidad del salón debe de ser mayor o igual a la cantidad de alumnos permitidos dentro de una clase.

RO7. Los dictados de clase perteneciente a un mismo grupo de cada ciclo académico no deben de solaparse entre ellos, ni en duración ni ubicación física, una vez asignada la hora de inicio de éstos y el salón correspondiente.

RO8. Las clases que son de teoría o de práctica o de laboratorio deben asignarse a salones que cumplan con las características necesarias para el dictado de la clase.

1.2.2 Restricciones Deseables

RD1. Las clases deberían dictarse dentro de una franja horaria, es decir, se debe asignar un intervalo de tiempo por día en donde la clase se pueda dictar de acuerdo al ciclo académico al cual pertenece.

RD2. Para las clases que pertenecen a un mismo ciclo académico, se debe limitar la cantidad de horas de dictado asignados en un mismo día, con un máximo de 6 periodos consecutivos.

RD3. Para las clases que pertenecen a un ciclo se debe limitar la cantidad de horas de dictado asignados en un mismo día, con un máximo de 9 periodos.

RD4. Los intervalos de tiempo entre dictados de clases que pertenecen a un grupo de un mismo ciclo deben minimizarse.

1.3 Objetivos

1.3.1 Objetivo Principal

Diseñar un sistema para generar en forma automática los horarios para la enseñanza en las universidades considerando los recursos disponibles que a diferencia de los horarios elaborados actualmente permitan cumplir con todas las restricciones obligatorias y en lo posible cumplir con la mayor cantidad de restricciones deseables.

1.3.2 Objetivos Secundarios

Los objetivos secundarios son los siguientes:

OS1. Revisar y evaluar algoritmos para generación automática de horarios.

OS2. Identificar las restricciones del problema.

OS3. Diseñar un algoritmo que proporcione una solución factible para la elaboración de horarios en base a la mejor técnica seleccionada de la revisión de la literatura.

OS4. Implementar un sistema que permita satisfacer la mayor cantidad de restricciones posibles.

OS5. Documentar los resultados obtenidos con este algoritmo computacional.

1.4 Justificación

Con este trabajo se pretende reducir el tiempo y la ocurrencia de errores en la generación manual de los horarios de clase por parte de las autoridades responsables, lo cual crea problemas como cruces de horarios, alumnos rezagados en uno o más semestres si es que los grupos aperturados no poseen cupos disponibles o cuando un alumno matriculado en un horario con un determinado docente, se encuentra posteriormente con que el horario cambió o que el dictado del curso estará a cargo de otro docente, generando insatisfacción entre docentes y estudiantes. Por otro lado en el plano económico, según la SUNEDU [42], hay 1491 direcciones de escuela, ver Figura 1.3, que tienen que publicar los horarios para todos los ciclos académicos impartidos durante 1 año, además el tiempo promedio requerido para generar el horario es de 1 mes en promedio (8 horas por día laboral), en total se tiene 357 840 horas de trabajo humano

que se desperdicia para la elaboración de los horarios para cada ciclo y estas personas son profesionales encargados de la jefatura de la dirección y reciben un pago por sus servicios el cual es remunerada por la Universidad, asumiendo que el encargado de ejercer dicha labor percibe un salario promedio de 2000 dólares mensuales y se realiza un horario por cada Dirección de Escuela tenemos un promedio de 2982000 dólares destinados a esta labor.



Figura 1.4 Número de Facultades y Carreras Profesionales que ofrecen las Universidades Peruanas [42].

La tasa de crecimiento anual de las Universidades Peruanas es 4.2, como se vio en la Figura 1.2, esto quiere decir que habrá más encargados de elaborar horarios para el dictado de clases en diferentes Facultades y que necesitarán hacer en menor tiempo posible este proceso. En la Figura 1.5 se puede observar el número de universidades creadas por año según tipo de universidad.

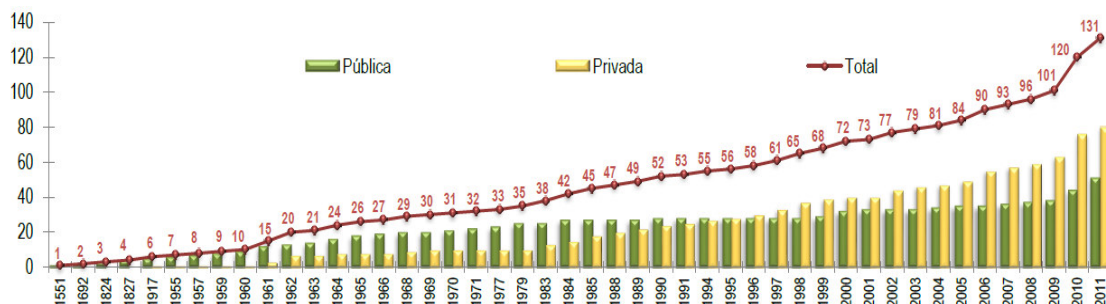


Figura 1.5 Número de Universidades creadas según tipo de universidad 1551-2011 [42].

El docente puede ser contratado en una o más Universidades Privadas y a la vez ser ordinario en una Universidad Pública o viceversa pero en la actualidad la plana docente de estas universidades el 79.7% son contratados, ver Figura 1.6, además los docentes pueden cambiar de categoría, esto indica que la disponibilidad de los docentes es variable y se requiere cumplir con los docentes asignándoles grupos en sus horas disponibles.

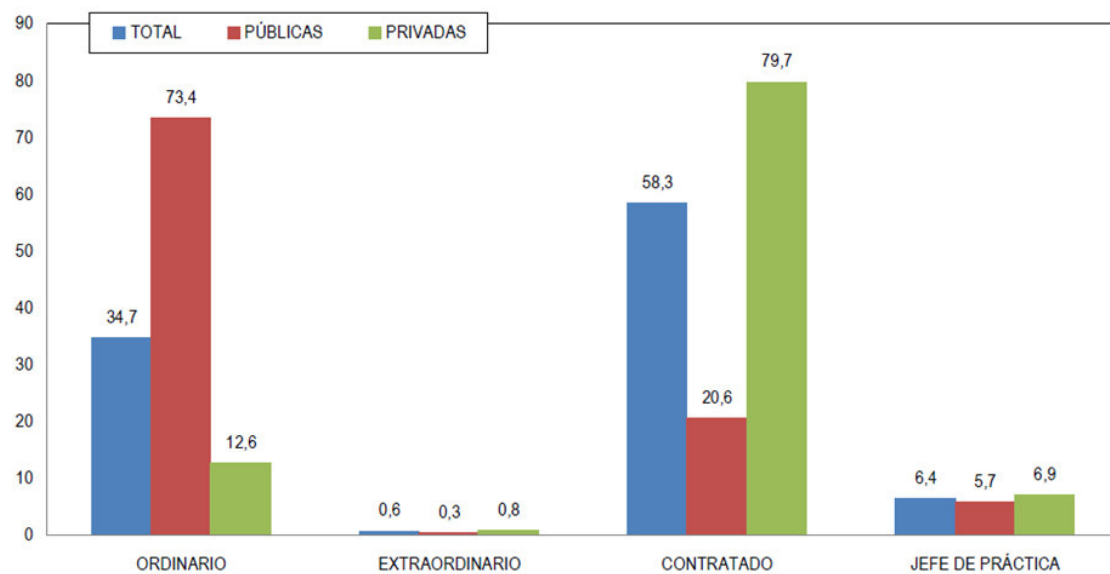


Figura 1.6 Porcentaje de Docentes Universitarios, por condición laboral según tipo de Universidad [41].

Este tipo de problema es intratable por lo cual no se puede determinar manualmente una solución óptima.

Para la Acreditación Universitaria que solicita el CONEAU [43] se requiere automatizar los procesos manuales que las Facultades tienen actualmente, una de ellas es la de elaborar los horarios.

El implantar una solución automatizada donde se reduzca el tiempo y la frecuencia de errores en la asignación manual de los horarios de clases permitirá satisfacer al máximo las necesidades y requerimientos de los usuarios (alumnos, docentes y directores). Esto implicará un gran avance tecnológico en las Facultades de las Universidades del Perú.

1.5 Alcances

- Se hará la revisión de todo lo que existe sobre el problema de horarios.
- El sistema será diseñado y probado en la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos.
- El sistema debe de abarcar los diferentes planes curriculares de la facultad.
- El sistema debe permitir fijar dictados de clases en un determinado periodo ya que hay docentes de otras facultades que indican un horario específico para dictar.
- El sistema podrá ser implementado en las Facultades que posean las restricciones anteriormente mencionadas en los puntos 1.2.1 y 1.2.2.

1.6 Propuesta

La solución propuesta es un sistema de generación de horarios basada en el algoritmo genético propuesto por Shengxiang Yang, Member y Sadaf Naseem Jat [1], el cual fue adaptado según nuestros requerimientos.

El sistema propuesto tiene el siguiente Patrón de Arquitectura:

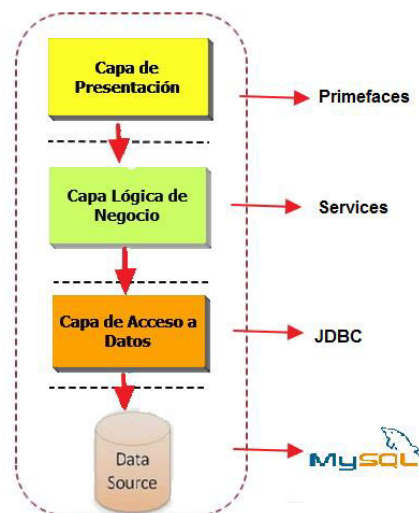


Figura 1.7 Patrón de Arquitectura del Sistema Propuesto [Fuente: Elaboración Propia].

1.7 Organización de la Tesis

La presente investigación consta de 7 capítulos, los cuales se detallan a continuación.

El capítulo uno presenta una introducción que muestra la problemática que ha motivado a realizar dicha investigación, así también como el objetivo principal, los objetivos secundarios, la justificación, los alcances, la propuesta y la estructura del documento.

El capítulo dos tiene como objetivo llevar a cabo una revisión de la literatura existente en el problema de generación de horarios de cursos en universidades, además de mostrar una comparación de los resultados en las diferentes instancias de los algoritmos.

El capítulo tres explica el algoritmo seleccionado y la parte seleccionada del algoritmo que se va a utilizar en la solución propuesta.

El capítulo cuatro explica la adaptación del algoritmo seleccionado en un modelo general y particular asociado al caso de estudio, se definen las estructuras y variables para validar que no se violen las restricciones obligatorias y minimizar las violaciones de las restricciones deseables.

En el capítulo cinco se dará a conocer el desarrollo del software ideado para la solución del problema tratado en este trabajo. Se mostrarán los requerimientos mínimos de hardware y software para su correcto funcionamiento, luego se detallará el modelo de la base de datos y por último los módulos funcionales del sistema.

En el capítulo seis se realiza el estudio experimental de la solución del problema de programación de horarios mediante el algoritmo genético, se explicará cómo se organizan las Facultades, se realizará la prueba de análisis de sensibilidad de los parámetros y la comparación con un caso de estudio.

En el capítulo siete se presentan las conclusiones a las que se llegaron con el estudio. Se responde los objetivos generales y específicos de la tesis; y se enuncian los trabajos futuros.

Capítulo 2: Revisión de algoritmos para el UCTP

En este capítulo hacemos una revisión de la literatura sobre algoritmos para el problema de generaciones de horarios de cursos universitarios, luego mostramos las propuestas de los autores con los resultados obtenidos en sus pruebas experimentales y finalmente hacemos un benchmarking de los resultados de los algoritmos probados con el dataset propuesto por Socha et al [6].

2.1 Algoritmos para generación de horarios a lo largo de la historia

Una breve historia sobre algoritmos para el problema de horarios ha sido escrita por Shengxiang Yang et al [1], quienes manifiestan que el primer trabajo para el problema de horarios fue escrito por Gotlieb en 1962 [9]. Además manifiesta que existen varios algoritmos para resolver problemas de horarios. El primer conjunto de algoritmos están basados en métodos heurísticos de coloración de grafos. Estos algoritmos muestran una gran eficiencia en pequeños casos de problemas de horarios, pero no son eficaces en casos de gran tamaño. Más tarde, los métodos meta-heurísticos se introdujeron para resolver los problemas de horarios con el principal fin de reducir al mínimo el número de violaciones de restricciones blandas en lugar de cumplir con las restricciones duras como Algoritmos Genéticos (GAs), Recodido Simulado (SA) [11], Búsqueda Tabú (TS) [13], entre otros algoritmos. De acuerdo a Mohammed Azmi Al-Betar et al [2], recientemente los más genéricos algoritmos hyperheurísticos se han aplicado con éxito para resolver el problema de horarios de cursos universitarios. Una revisión completa y recientes direcciones de investigación pueden encontrarse en Hamed Babaeia et al [27].

Quet al. [28] and Lewis [29].

Según Shengxiang Yang, entre las meta-heurísticas utilizadas en horarios están los algoritmos de optimización en una etapa y los algoritmos de optimización de dos etapas. Mientras que en los algoritmos de optimización de una etapa, intenta satisfacer las restricciones duras y blandas al mismo tiempo, en los algoritmos de optimización de dos etapas, la satisfacción de las restricciones blandas se intenta sólo cuando un horario factible se ha encontrado. Generalmente hablando hay dos tipos de algoritmos meta -

heurísticos [10]: algoritmos basados en área local y algoritmos basados en la población. Cada tipo tiene algunas ventajas y desventajas. Los algoritmos basados en área local incluyen SA [11], búsqueda local muy grande e iterativa [12], TS [13], el gran diluvio (GD) [26] y muchos más. Por lo general, los algoritmos basados en área local se centran en la explotación en lugar de la exploración, lo que significa que se mueven en una dirección sin realizar un análisis más amplio del espacio de búsqueda. Los algoritmos basados en la población empiezan con un número de soluciones y hay que afinarlos para obtener la solución óptima global en el conjunto de espacio de búsqueda, y por lo tanto, son algoritmos basados en el área global. Los algoritmos basados en la población que se utilizan normalmente para hacer frente a los problemas de horarios incluyen algoritmos evolutivos [50], la optimización de enjambre de partículas [14], la optimización de colonia de hormigas [15], el algoritmo de búsqueda armónica [4], el sistema inmune artificial [16], etc.

En los últimos años, varios investigadores han utilizado algoritmos genéticos para resolver el UCTP, para más información sobre algoritmos genéticos en [38]. Se mejoró el rendimiento de los algoritmos genéticos utilizando operadores genéticos modificados, los operadores heurísticos y las técnicas de búsqueda local. La calidad de una solución producida por algoritmos basados en la población no puede ser superior al de algoritmos basados en área local, debido principalmente al hecho de que los algoritmos basados en la población están más interesados en la exploración que en la explotación [10]. Los algoritmos basados en la población exploran soluciones en el conjunto de espacio de búsqueda, sin centrarse en los individuos con buena aptitud dentro de una población. Por otra parte, los algoritmos basados en la población pueden experimentar convergencia prematura, que puede llevar a que sean atrapados en óptimos locales. Los algoritmos basados en la población tienen otro inconveniente de requerir más tiempo [17]. Sin embargo, los algoritmos evolutivos tienen varias ventajas en comparación con otras técnicas de optimización [18]. Por ejemplo, los algoritmos genéticos pueden realizar una búsqueda multidireccional usando un conjunto de soluciones candidatos [1].

En la Figura 2.1 se muestran algunos de los algoritmos usados para resolver el problema de horarios de cursos universitarios a lo largo de la historia.

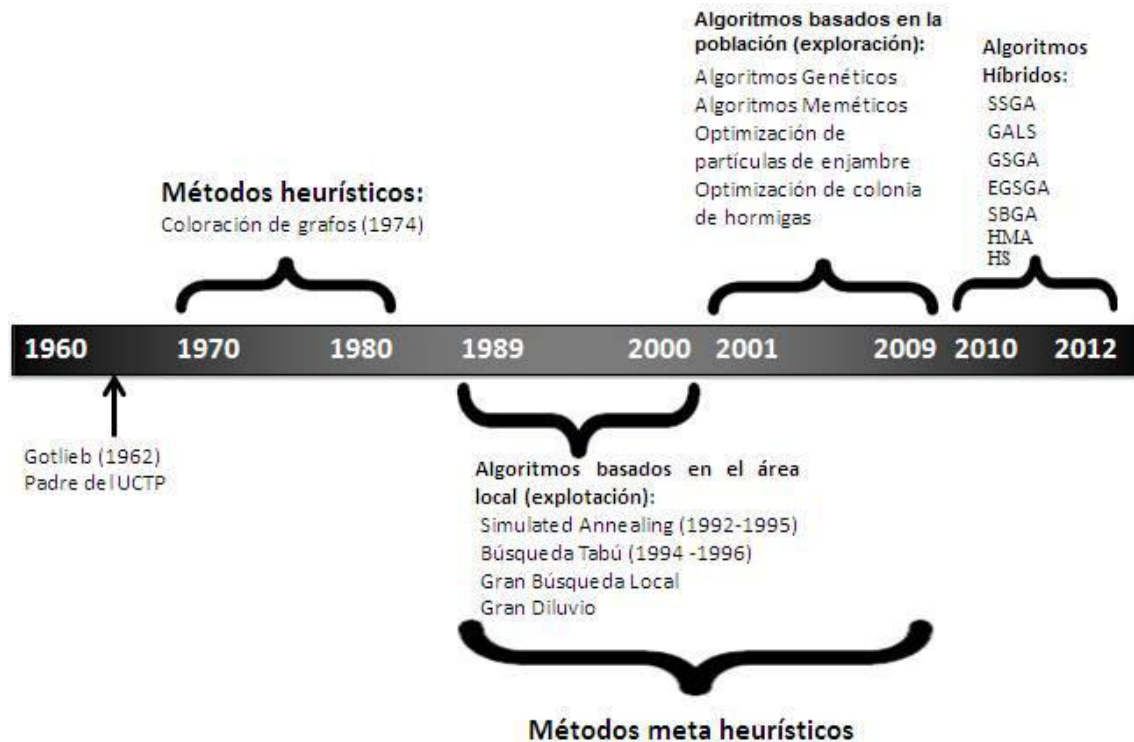


Figura 2.1 Línea de tiempo de Algoritmos propuestos para solucionar el UCTP [Fuente: Elaboración Propia].

2.2 Algoritmos para Generación de Horarios

2.2.1 Un algoritmo genético extendido con búsqueda local y búsqueda guiada (EGSGA)

Este trabajo fue realizado por los autores Shengxiang Yang, Member y Sadaf Naseem Jat [1], en el año 2011, quienes desearon combinar las buenas propiedades de algoritmos basados búsqueda de área local y global para resolver el UCTP. Trataron de hacer un balance entre la capacidad de exploración, mejoría global, de algoritmos genéticos (AGs) y la capacidad de explotación, mejora local, de la búsqueda local (LS).

Los autores propusieron una versión extendida de GSGA, denotado EGSGA, para el UCTP. En el EGSGA, un nuevo algoritmo de búsqueda local denotado como LS2 es añadido al algoritmo GSGA. El algoritmo general (GSGA) es descrito a continuación:

En el algoritmo genético con búsqueda guiada y búsqueda local (GSGA), primero se inicializa la población mediante la creación de cada individuo al azar a través de la

asignación de un intervalo de tiempo aleatorio para cada evento de acuerdo con una distribución uniforme y aplicando el algoritmo de coincidencia de asignar una aula para el evento. Entonces, el algoritmo LS1 se aplica a cada miembro de la población inicial usando las tres estructuras vecinales. Después de la inicialización de la población, se construye una estructura de datos llamada MEM, que almacena una lista de pares de aula e intervalo de tiempo (r, t) para cada evento que no tiene penalización (es decir, no ha habido violación de restricciones duras y blandas en este caso) de individuos seleccionados de la población. Después de eso, el MEM puede ser utilizado para guiar la generación de descendientes para las siguientes generaciones. La estructura de datos MEM se reconstruye periódicamente, por ejemplo, cada τ generaciones. En cada generación se genera un hijo ya sea mediante el uso de MEM o mediante la aplicación del operador de cruce, en función de una probabilidad γ . Después de esto el hijo va a ser mutado por un operador de mutación con una probabilidad p_m seguido por el algoritmo de búsqueda local LS1. Por último, el peor miembro de la población se sustituye por el individuo recién generado. Esta iteración continúa hasta que se alcanza una condición de término, por ejemplo, cuando se alcanza un límite de tiempo preestablecido. La Figura 2.2 muestra el algoritmo base (GSGA).

Algorithm 3 Guided Search Genetic Algorithm (GSGA)

```

1: input : A problem instance  $I$ 
2: randomly initialize a population of solutions
3: evaluate the individuals in the population
4: apply the local search scheme LS1 to each individual of
   the population
5: set the generation counter  $g := 0$ 
6: while the termination condition is not reached do
7:   if  $(g \bmod \tau) == 0$  then
8:     apply ConstructMEM() to construct MEM
9:   end if
10:  create a child using GuidedSolutionConstruction()
    or Crossover() with a probability  $\gamma$ 
11:  apply mutation to the child with a probability  $P_m$ 
12:  evaluate the child solution
13:  apply local search LS1 to the child
14:  replace the worst member of the population by the child
15:   $g := g + 1$ 
16: end while
17: output : The best solution  $s_{best}$  achieved for  $I$ 

```

Figura 2.2 Algoritmo Genético con Búsqueda Guiada (GSGA) [1].

A continuación describimos el aporte del autor que es el algoritmo de búsqueda local LS2 el cual selecciona aleatoriamente un porcentaje preestablecido de intervalo de tiempo del total de intervalos de tiempo de T . A continuación, se calcula la penalidad de cada intervalo de tiempo seleccionado y elige el peor intervalo de tiempo w_t que tiene el mayor valor de penalidad de LS. Después de tomar el peor intervalo de tiempo, LS2 aplica la estructura vecinal N1 para cada evento de w_t y comprueba el valor de penalización de cada caso, antes y después de aplicar el movimiento. Si todos los movimientos en w_t juntos reducen las violaciones de restricciones duras y / o blandas, entonces se aplican los movimientos, de lo contrario, no se aplican los movimientos.

En la figura Fig. 2.3 se muestra el algoritmo de búsqueda LS2.

Algorithm 7 Local Search Scheme 2 (LS2)

```

1: input : Individual  $I$  after LS1 is applied
2:  $S :=$  randomly select a preset percentage of time slots from the
   total time slots of  $T$ 
3: for each time slot  $t_i \in S$  do
4:   for each event  $j$  in time slot  $t_i$  do
5:     calculate the penalty value of event  $j$ 
6:   end for
7:   sum the total penalty value of events in time slot  $t_i$ 
8: end for
9: select the time slot  $w_t$  with the biggest penalty value from  $S$ 
10: for each event  $i$  in  $w_t$  do
11:   calculate a move of event  $i$  in the neighbourhood structure N1
12:   apply the matching algorithm to the time slots affected by the
      move
13:   compute the penalty of event  $i$  and delta evaluate the result
14: end for
15: if all the moves together reduce hard or soft constraint violations
      then
16:   apply the moves
17: else
18:   delete the moves
19: end if
20: output : A possibly improved individual  $I$ 

```

Figura 2.3 Algoritmo de Búsqueda Local (LS2) [1].

En los resultados experimentales se hicieron comparaciones con otros algoritmos (Randomized Iterative Improvement Algorithm (RIIA) [12], Nonlinear Great Deluge (NLGD) [19], GA With LS (GAWLS) [20], Hybrid EA (HEA) [21], Graph-Based Hyperheuristic (GBHH) [22], Tabu Search Hyperheuristics (THHS) [23], LS [15], GA [24], AA [6], Fuzzy Algorithm (FA) [25], con los tres tamaños de instancia (pequeño, mediano y grande) y arrojaron que el algoritmo EGSGA es mejor en el 90.90% de los casos como se muestra en la Tabla 2.1.

UCTP	EGSGA		RIIA Best	NLGD Best	GAWLS Best	HEA Best	GBHH Best	THHS Best	LS Med	GA Best	AA Med	FA Best
	Best	Med										
S1	0	0	0	3	2	0	6	1	8	0	1	10
S2	0	0	0	4	4	0	7	2	11	3	3	9
S3	0	0	0	6	2	0	3	0	8	0	1	7
S4	0	0	0	6	0	0	3	1	7	0	1	17
S5	0	0	0	0	4	0	4	0	5	0	0	7
M1	139	143	242	140	254	221	372	146	199	280	195	243
M2	92	96.5	161	130	258	147	419	173	202.5	188	184	325
M3	122	124	265	189	251	246	359	267	77.5%ln	249	248	249
M4	98	101	181	112	321	165	348	169	177.5	247	164.5	285
M5	116	119.5	151	141	276	135	171	303	100%ln	232	219.5	132
L	615	622.5	100%ln	876	1027	529	1068	80%ln	100%ln	100%ln	851.5	1138

Tabla 2.1 Comparación del número de violaciones de restricciones blandas del algoritmo EGSGA con otros algoritmos de la literatura [1].

2.2.2 EL algoritmo Multiswap

El algoritmo MultiSwap fue propuesto por Mohammed Azmi AI-Betar, Ahamad Tajudin Khader, Osama Muslihl [2], en el año 2012, quienes consideraron que las asignaciones de aulas son esenciales en el contexto del problema de horarios de cursos universitarios, además nos dice que los trabajos anteriores estaban más preocupados por las operaciones de intervalos de tiempo (cualquier operación realizada entre un evento y el intervalo de tiempo) y pasar por alto las operaciones de aulas (cualquier operación realizada entre un evento y un aula en el mismo intervalo de tiempo).

En la Figura 2.4 se muestra el esquema del algoritmo propuesto, en donde se utiliza un paradigma de dos etapas en donde en la primera etapa, el mayor grado ponderado primero (LWD), el algoritmo backtracking, y el algoritmo de MultiSwap se combinan para encontrar un horario factible. En la segunda etapa, el algoritmo de búsqueda local y el algoritmo MultiSwap se utilizan para mejorar la calidad del horario.

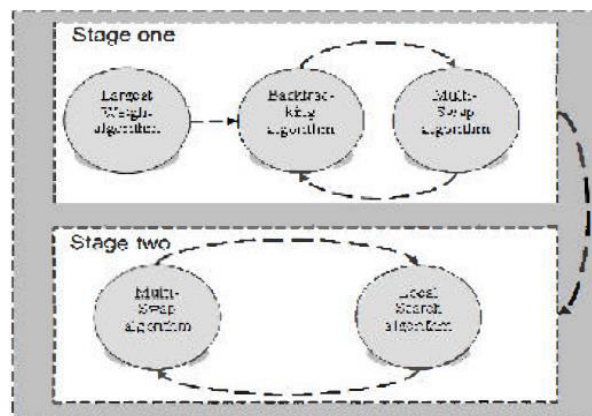


Figura 2.4 Esquema del algoritmo propuesto [2].

El algoritmo MultiSwap es propuesto por los Mohammed Azmi AI-Betar et al. [2], para hacer frente a las operaciones de las aulas, en donde, para cada intervalo de tiempo, los eventos son mezclados aleatoriamente a las diferentes aulas validas dentro del mismo intervalo de tiempo. El algoritmo MultiSwap es una idea simple de las operaciones de aulas, fácil de incorporar con otros algoritmos, y tiene un efecto insignificante en el tiempo computacional. El pseudo-código del algoritmo MultiSwap es el siguiente:

De acuerdo con lo descrito por los autores, se puede comenzar con un horario completo (es decir, se han asignado todos los eventos al horario) o incompleto, en donde se toma los intervalos de tiempo consecutivos donde P es el conjunto de intervalos de tiempo. Los eventos de cada intervalo de tiempo p son mezcladas aleatoriamente a las aulas adecuadas. La salida de este algoritmo es otro horario completo o incompleto, pero con arreglos diferentes en las aulas para el evento en el mismo intervalo de tiempo.

En la Figura 2.5 se muestra el algoritmo MultiSwap que ha cubierto dos operaciones del aula:

- (i) Mover un evento de un aula a otra vacía dentro del mismo intervalo de tiempo.
- (ii) Permutar eventos entre dos aulas en el mismo intervalo de tiempo.

```

Input: complete or incomplete course timetable (A)
1: begin
2: for all  $p \in P$  //P is the timeslot set
3: for all  $i \in R$  //R is the room set
4: for all  $j \in R$   $j \neq i$ 
5: begin
6: if place  $a_{ip}$  contains event  $e$  and place  $a_{jp}$  empty then
7: if room  $i$  is appreciate room for event  $e$  then
8: move event  $e$  to place  $a_{jp}$ 
9: if place  $a_{ip}$  contains event  $e$  and place  $a_{jp}$  contains event  $q$  then
10: if room  $i$  appreciates to  $a_{jp}$  and room  $j$  appreciates to  $a_{ip}$  then
11: swap the places ( $a_{ip}$ ,  $a_{jp}$ ) of event  $e$  and  $q$ 
12: end
13: end
Output: course timetable with same quality buy different rooms' arrange

```

Figura 2.5 El pseudo-código del algoritmo MultiSwap [2].

Los resultados experimentales en donde se hicieron comparaciones con otras técnicas y se obtuvo que el Algoritmo Multiswap con Búsqueda Local (MSLS) resulto ser el mejor en el 18.18% de los casos como se muestra en la Tabla 2.2.

Dataset	Initialization	MSLS (best)	RRLS (Avg.)	MMAS (Avg.)	THH (best)	VNS (best)	FMHO (best)	GHH (best)	RII (best)	HEA (best)	HSA (best)
small1	196	2	8	1	1	1	10	6	0	0	5
Small2	194	4	11	3	2	2	9	7	0	0	3
Small3	231	2	8	1	0	0	7	3	0	0	2
Small4	252	2	7	1	1	1	17	3	0	0	3
Small5	262	0	5	0	0	0	7	4	0	0	1
Medium1	822	174	199	195	146	317	243	372	242	221	316
Medium2	789	184	202	184	173	313	225	419	161	147	243
Medium3	786	188	77.5% inf.	248	267	357	249	359	265	246	255
Medium4	772	180	178	164.5	169	247	285	348	181	165	235
Medium5	783	132	100% inf.	219.5	303	292	132	171	151	130	215
Large	1608	994	100% inf.	851.5	1166	100%inf.	1138	1068	100% inf.	529	100% inf

Tabla 2.2 Comparación del número de violaciones de restricciones blandas Multiswap con otros algoritmos de la literatura [2].

2.2.3 Uso de las estructuras vecinales múltiples dentro de un enfoque memético basado en Tabú a problemas de horarios universitarios

Este trabajo fue realizado por Salwani Abdullah, Hamza Turabieh [3], en el año 2012, quienes se basaron en investigaciones de la literatura que establecían que la mayoría de los métodos metaheurísticos utilizan las estructuras vecinales en sus algoritmos de búsqueda. Según Salwani Abdullah et al, la investigación también ha demostrado que la hibridación de métodos metaheurísticos a menudo se desempeña mejor que los algoritmos individuales, ya que se benefician de las ventajas de ambas (o más) técnicas. Esto conduce a la investigación de un método híbrido que utilice grandes estructuras vecinales para resolver el problema de horarios.

Los autores propusieron un algoritmo memético basado en Tabú, que es un algoritmo híbrido de un algoritmo genético con un algoritmo de búsqueda tabú como un algoritmo mejorado para problemas de horarios de cursos universitarios. Este algoritmo se emplea con conjunto de estructuras vecinales durante el proceso de búsqueda con el objetivo de obtener mejoras significativas en la calidad de la solución. La secuencia de las estructuras vecinales se ha considerado para entender su efecto sobre el espacio de búsqueda. El concepto de una lista tabú está incrustado para controlar la selección de estructuras vecinales que no son dependientes de los dominios de problemas durante el proceso de optimización después de los operadores de cruce y mutación se aplican a las soluciones seleccionadas del conjunto de la población. El algoritmo penalizará a las estructuras vecinales que son incapaces de generar mejores soluciones. El algoritmo comienza mediante la creación de una población inicial, se selecciona la mejor solución

de la población, S_{best} . Una lista Tabú con el tamaño $TabuSize$ es creada para evitar que ineficaces estructuras vecinales sean seleccionadas en la siguiente iteración y proporcionar una mejor oportunidad para que las estructuras vecinales restantes puedan ser exploradas. Una variable $CanSelect$ permite al algoritmo controlar la selección de una estructura vecinal. Por ejemplo, si una estructura vecinal, Nbs , exhibe un buen rendimiento el algoritmo, Nbs se seguirá utilizando en la siguiente iteración hasta que una no buena solución se pueda obtener. En este caso, Nbs será empujado dentro de la Lista de Tabú FIFO, T_{List} . A este movimiento no se le permite ser parte del proceso de búsqueda para un cierto número de iteraciones, que es igual al tamaño de la lista tabú. En un bucle do-while, dos soluciones son seleccionados al azar, S_1 y S_2 . Los operadores de cruce y la mutación serán aplicados a S_1 y S_2 para obtener S_1' y S_2' . Una estructura vecinal, Nbs , se selecciona al azar y se aplica a S_1' y S_2' para obtener S_1'' y S_2'' . La mejor solución entre los S_1 , S_2 , S_1' , S_2' , S_1'' y S_2'' es elegido y asignado a un solución actual S^* . Si S^* es mejor que la mejor solución S_{best} , entonces, S^* es aceptada. De lo contrario, se agregará la estructura vecinal Nbs a T_{List} . Los miembros de las poblaciones serán actualizados mediante la eliminación de la peor solución y la inserción de la nueva solución obtenida del proceso de búsqueda, mientras que se mantiene el tamaño de la población a ser utilizado en la próxima generación. El proceso se repite hasta que se cumple el criterio de término. En la Figura 2.6 se muestra el pseudo-código del algoritmo Memético basado en Tabú.

```

do while ( $i < \text{population size}$ )
    Generate random solutions,  $S_i$ ;
end while
Set the length of the tabu,  $TabuSize$ ;
Choose a best solution from the population,  $S_{best}$ ;
Create an empty tabu list with  $TabuSize$ ,  $T_{List}$ ;
Set  $CanSelect \leftarrow True$ 
do while (not termination criterion)
    Select two parents from the population using a roulette wheel selection,
     $S_1$  and  $S_2$ ;
    Apply crossover and mutation operators on  $S_1$  and  $S_2$  to produce  $S_1'$  and
     $S_2'$ ;
    if  $CanSelect == True$ 
        Select a neighbourhood structure, which is not in  $T_{List}$ , called  $Nbs$ ;
    end if
    Apply  $Nbs$  on  $S_1'$  and  $S_2'$  to produce  $S_1''$  and  $S_2''$ ;
    Obtain a minimum solution penalty from  $S_1$ ,  $S_2$ ,  $S_1'$  and  $S_2'$ , called
    current solution  $S^*$ ;
    if ( $S^* < S_{best}$ )
         $S_{best} \leftarrow S^*$ ;
         $CanSelect \leftarrow False$ ;
    else
        Push  $Nbs$  to  $T_{List}$ ;
         $CanSelect \leftarrow True$ ;
    end if
    Update the sorted population by inserting and removing the best and
    worst solutions, respectively, while maintaining the size of the
    population;
end while
Output the best solution,  $S_{best}$ ;

```

Figura 2.6 Algoritmo Memético basado en Tabú [3].

Los resultados experimentales fueron comparados en 8 instancias con los resultados de los mejores algoritmos del concurso de la Segunda Competencia Internacional de Horarios (ITC2007), en donde fue mejor en todas las instancias, pero no fue mejor en ninguna instancia comparados con el mejor resultado del algoritmo conocido utilizado para resolver cada instancia del problema, por lo tanto no se obtuvieron resultados prometedores.

2.2.4 Un nuevo algoritmo de búsqueda armónica

Este algoritmo fue propuesto por Mohammed Azmi Al-Betar; Ahamad Tajudin Khader; Taufiq Abdul Gani [4], en el año 2012, quienes basados en revisiones de la literatura sugirieron que la mejor forma de diseñar un algoritmo de programación de horarios es proveer un buen equilibrio entre los algoritmos basados en la localidad y los algoritmos basados en la población para alcanzar un adecuado equilibrio entre la exploración (mejoría global) y explotación (mejora local).

En la Figura 2.7 se muestra el esquema del algoritmo propuesto:

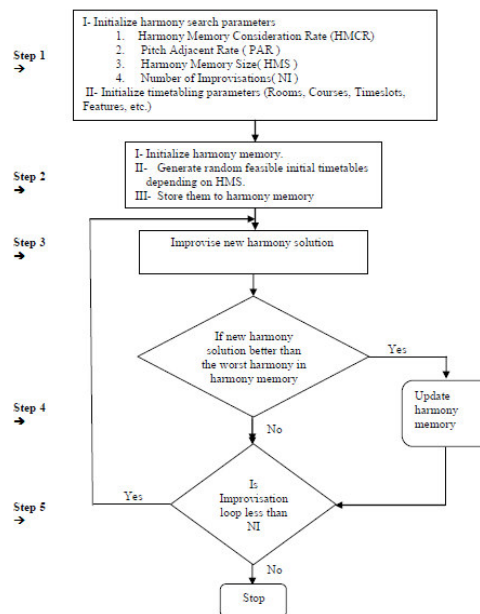


Figura 2.7 Algoritmo de búsqueda armónica [4].

Paso1: Inicializa los parámetros

1 HMCR: Tasa de consideración de la memoria de la armonía similar a la tasa de cruce del algoritmo genético.

2. PAR: Tasa de ajuste que juega un papel importante en la determinación del número de cursos que van a ser movidos a otra posición o intercambiados con otros cursos como las estructuras vecinales de búsquedas locales.

3. HMS: Tamaño de la memoria de la armonía es similar al tamaño de la población.

4. NI: Número de Improvisaciones es similar al número de iteraciones en algoritmos de optimización.

Paso 2: Inicialización de la memoria de la armonía.

El algoritmo de backtracking [30] y el algoritmo MultiSwap son utilizados para generar soluciones aleatorias de la armonía igual a HMS después de asignar los cursos por la heurística del grado ponderado más grande primero [31].

Paso 3: Improvisar nueva solución armónica

Una nueva solución factible es generada basada en 3 operadores:

Consideraciones de Memoria: La nueva posición de un curso es escogido de las posiciones históricas de horarios almacenados en HM con una probabilidad HMCR.

Consideraciones Aleatorias: Los $(1-HMCR)$ cursos que no son asignados con el operador anterior serán asignados de manera aleatoria.

Ajustes de tono: El ajuste de tono trabaja similar a estructuras vecinales de algoritmos basados en la localidad que es concerniente con la explotación mientras el operador Consideración de Memoria es concerniente con la exploración.

Paso 4: Actualización de la memoria armónica.

Si el valor de la función de aptitud de la nueva solución armónica es mejor que el peor valor de la función de aptitud en HM, incluimos la nueva solución en HM y excluimos la peor solución de HM.

Paso 5: En el algoritmo de búsqueda armónica (HS), se va a repetir los pasos 3 y 4 hasta que el máximo número de iteraciones determinado por los parámetros NI sea alcanzado.

En los resultados experimentales se hicieron comparaciones con otros algoritmos (Randomized Iterative Improvement (RII), Random Restart Local Search (RRLS), GA MAX-MIN Ant System (MMAS), Variable NeighborhoodSearch (VNS), Graph-based Hyper Heuristic (GHH), Fuzzy Multiple Heuristic Ordering (FMHO), Hybrid

Evolutionary Approach (HEA), Tabu-search Hyper-Heuristic (THH) con los tres tamaños de instancia (pequeño, mediano y grande) donde se observa que los resultados no son tan buenos como los resultados de los otros algoritmos dentro de las mismas instancia de datos además muestran que el algoritmo HS es capaz de encontrar soluciones viables solo para las instancias de datos pequeñas y medianas como se muestra en la Tabla 2.3:

Datasets	HS (best)	RJ1 (best)	RRLS (Avg)	MMAS (Avg)	VNS (best)	GHH (best)	FMHO (best)	HEA (best)	THH (best)
small1	5	0	8	1	1	6	10	0	1
Small2	3	0	11	3	2	7	9	0	2
Small3	2	0	8	1	0	3	7	0	0
Small4	3	0	7	1	1	3	17	0	1
Small5	1	0	5	0	0	4	7	0	0
Medium1	316	242	199	195	317	372	243	221	146
Medium2	243	161	202.5	184	313	419	225	147	173
Medium3	255	265	77.5% inf.	248	357	359	249	246	267
Medium4	235	181	177.5	164.5	247	348	285	165	169
Medium5	215	151	100% inf.	219.5	292	171	132	130	303
Large	100% inf	100% inf.	100% inf.	851.5	100% inf.	1068	1138	529	80% inf. 1166

Tabla 2.3 Comparación del número de violaciones de restricciones blandas del algoritmo HS con otros algoritmos de la literatura [4].

2.2.5 Un enfoque meta-heurístico híbrido

El enfoque meta-heurístico híbrido fue propuesto por Salwani Abdullah, Hamza Turabieh, Barry McCollum y Paul MacMullan [5], en el año 2012, en donde proponen una nueva meta-heurística que combina un mecanismo de electromagnetismo(EM) [32] y el algoritmo del Gran Diluvio(GD) que fue introducido por primera vez por Dueck [33], para resolver el problema de horarios de cursos universitarios(UCTP). EM es un algoritmo estocástico de optimización global basado en la población que se basa en la teoría de la física, la simulación de atracción y repulsión de los puntos de muestreo para avanzar hacia la optimalidad. GD es un procedimiento de búsqueda local que permite a las peores soluciones ser aceptado en base a algún límite superior dado o “nivel”. La fuerza dinámica calculada a partir del mecanismo de atracción-repulsión se utiliza como una tasa decreciente a actualizar el “nivel” en el proceso de búsqueda.

En la Figura 2.8 se muestra el esquema de general del algoritmo propuesto y en la Figura 2.9 muestra el algoritmo de tasa de descomposición de fuerza dinámica del algoritmo del gran diluvio.

En el paso 1 por cada población se mide la Calidad de la solución $Soli$ es llamado $f(Soli)$. Al principio, la mejor solución entre $Soli$ es establecido como $Solbest$. El nivel inicial se establece como $f(Soli)$.

En el paso 2 por cada iteración para cada población es calculado el Total de Fuerza (TF_i) basado en el algoritmo de mecanismo de electromagnetismo. Este valor (TF_i) se utiliza para calcular la calidad estimada, $EstimatedQuality$, y la tasa de decaimiento, β .

Seguidamente se aplica una estructura vecinal que asigna aleatoriamente cursos a un nuevo intervalo de tiempo válido. Esta medida generará una nueva solución, $Soli^*$. La calidad de la nueva solución se mide, $f(Soli^*)$. $Soli^*$ será aceptada si $f(Soli^*)$ es mejor que la mejor solución, $f(Solbest)$, o si $f(Soli^*)$ es inferior o igual al "nivel". Entonces, el "nivel" se reducirá por el valor de β . El proceso se repite y se detiene cuando se cumplen los criterios de terminación.

```

Step 1:      Initialization:
              Generate a population of initial solution as  $Soli$  where  $i = \text{size of the}$ 
              population;
              Calculate the initial penalty cost,  $f(Soli)$ ;
              Find the best solution among  $Soli$ , called  $Solbest$ ;
              Set total number of iterations,  $NumOfIt$ ;
              Set number of iteration in great deluge algorithm,  $NumGD$  (note that
               $NumGD$  is set to 1000 in our experiments)
              Set iteration  $\leftarrow 0$ ;
              Set iterationGD  $\leftarrow 0$ ;

Step 2:      Evaluation:
              do while (iteration <  $NumOfIt$ )
                for each population  $i$  where  $i = 1$  to population size;
                  Calculate total force,  $TF_i$ , based on electromagnetic-like
                  mechanism;
                  Set initial level: level  $\leftarrow f(Soli)$ ;
                  do while (iterationGD <  $NumGD$  ||  $f(Soli) < \text{level}$ )
                    Apply a dynamic force decay rate great deluge algorithm;
                  end do
                end for;
              end do

```

Figura 2.8. Pseudocódigo Genérico del algoritmo [5].

```

Calculate estimated quality of a solution,
 $EstimatedQuality = f(Soli) - TF_i$ ;
Calculate force decay rate,  $\beta =$ 
 $EstimatedQuality/NumOfIt$ ;
Define neighborhood of  $Soli$  by randomly assigning
course to a valid timeslot to generate a new solution
called  $Soli^*$ ;
Calculate  $f(Soli^*)$ ;
if ( $f(Soli^*) < f(Solbest)$ )
   $Soli \leftarrow Soli^*$ ;
   $Solbest \leftarrow Soli^*$ ;
else
  if ( $f(Soli^*) \leq \text{level}$ )
     $Soli \leftarrow Soli^*$ ;
  level = level -  $\beta$ ;

```

Figura 2.9. Pseudocódigo del algoritmo meta-heurístico híbrido [5].

En la Tabla 2.4 se muestra los resultados en términos de costo de penalidad con otros algoritmos disponibles en la literatura en 11 instancias. El término "x % Inf. " indica un porcentaje de ejecuciones que no lograron obtener soluciones factibles. Los mejores resultados se presentan en negrita. El algoritmo propuesto es capaz de encontrar horarios factibles para todos los casos.

Los algoritmos descritos en la Tabla 2.4 son descritos como sigue: M1: The genetic algorithm and local search [34], M2: The randomized iterative improvement algorithm [12], M3: The graph hyper heuristic [22], M4: The variable neighborhood search with tabu [35], M5: The hybrid evolutionary approach [21], M6: The extended great deluge [36], M7: The non linear great deluge [19], M8: The local search method [15], M9: The ant algorithm [6], M10: The fuzzy algorithm [25], M11: The evolutionary algorithm [24].

Dataset	Our method	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
<i>small1</i>	0	2	0	6	0	0	0	3	8	1	10	0
<i>small 2</i>	0	4	0	7	0	0	0	4	11	3	9	3
<i>small 3</i>	0	2	0	3	0	0	0	6	8	1	7	0
<i>small 4</i>	0	0	0	3	0	0	0	6	7	1	17	0
<i>small 5</i>	0	4	0	4	0	0	0	0	5	0	7	0
<i>medium1</i>	175	254	242	372	317	221	80	140	199	195	243	280
<i>medium2</i>	197	258	161	419	313	147	105	130	202.5	184	325	188
<i>medium3</i>	216	251	265	359	357	246	139	189	77.5% Inf	248	249	249
<i>medium4</i>	149	321	181	348	247	165	88	112	177.5	164.5	285	247
<i>medium5</i>	190	276	151	171	292	130	88	141	100% Inf	219.5	132	232
<i>large</i>	912	1026	100% Inf	1068	100% Inf	529	730	876	100% Inf	851.5	1138	100% Inf

Tabla 2.4 Comparación del número de violaciones de restricciones blandas del algoritmo meta heurístico híbrido con otros algoritmos de la literatura [5].

2.3 Benchmarking de algoritmos propuestos en la literatura

Para realizar la comparación de los resultados de los algoritmos presentados en la sección 2.2, los autores utilizaron como conjunto de datos de prueba el dataset de Socha et al [6], la cual se muestra en la Tabla 2.5. Este conjunto de datos está dividido en 3 categorías: pequeño, mediano y grande, se utiliza 11 instancias: 5 pequeñas, 5 medianas y 1 grande; los valores de los parámetros consisten de un número de eventos, aulas, características, número máximo de cursos para cada estudiante, número máximo de estudiantes para cada curso, promedio de características para cada aula y porcentaje de características usadas en cada categoría.

Class	Small	Medium	Large
Number of events	100	400	400
Number of rooms	5	10	10
Number of features	5	5	10
Approximate feature per room	3	3	5
Percent feature use	70	80	90
Number of students	80	200	400
Max events per student	20	20	20
Maximum student per event	20	50	100

Tabla 2.5 Características del dataset de Socha [6].

En la Tabla 2.6 se realiza una comparación de los resultados de los algoritmos presentados en la sección 2.2, el cual por cada instancia nos muestra el número de violaciones de restricciones blandas en donde podemos observar que el algoritmo EGSGA es mejor a los otros algoritmos en todos los casos. El término "x % Inf. " indica un porcentaje de ejecuciones que no lograron obtener soluciones factibles.

Dataset	EGSGA	Multiswap	Harmony Search	Hybrid metaheuristic approach
Small1	0	2	5	0
Small2	0	4	3	0
Small3	0	2	2	0
Small4	0	2	3	0
Small5	0	0	1	0
Medium1	139	174	316	175
Medium2	92	184	243	197
Medium3	122	188	255	216
Medium4	98	180	235	149
Medium5	116	132	215	190
Large	615	994	100% inf.	912

Tabla 2.6. Comparación del número de violaciones de restricciones blandas de los algoritmos presentados [1], [2], [4], [5].

Capítulo 3: Algoritmos Genéticos con Búsqueda

Guiada y Búsqueda Local de Shengxiang Yang y

Sadaf Naseem Jat

En este capítulo realizamos la descripción del problema en la que se basó el autor, seguidamente describimos el algoritmo GSGA (Algoritmos Genéticos con Búsqueda Guiada y Búsqueda Local) [7] propuesto por los mismos autores y su extensión EGSGA (Extensión de Algoritmos Genéticos con Búsqueda Guiada y Búsqueda Local) que es el algoritmo mejorado, aportando una nueva búsqueda local LS2 al algoritmo GSGA.

3.1 Descripción del Problema

En el problema de generación de horarios se asignan eventos (clases) en un periodo de tiempo y también se asignan un número de recursos (estudiantes y salones). Como es mencionado por Socha et al [6]. El problema de generación de horarios consiste en un conjunto de eventos $E = \{e_1, e_2, \dots, e_n\}$ a ser programado en 45 periodos $T = \{t_1, t_2, \dots, t_{45}\}$ (es decir 9 por cada día en 5 días a la semana), un conjunto de m salones disponibles $R = \{r_1, r_2, \dots, r_m\}$ en los que los eventos toman lugar, un conjunto de l estudiantes $S = \{s_1, s_2, \dots, s_k\}$ que atienden a los eventos, y un conjunto de l características disponibles $F = \{f_1, f_2, \dots, f_l\}$ que son satisfechos por los salones y requeridos por los eventos.

Además, las interrelaciones entre estos conjuntos están dadas por 5 matrices. La primera matriz $A_{k,n}$, llamado matriz Estudiante-Evento, muestra que eventos son atendidos por que estudiante. En $A_{k,n}$, el valor de $a_{i,j}$ es 1 si el estudiante $i \in S$ debería asistir al evento $j \in E$; de lo contrario, el valor es 0. La segunda matriz $B_{n,n}$, llamado matriz Evento-Conflicto, indica si dos eventos se pueden programar en el mismo periodo o no. Ayuda a identificar rápidamente los eventos que puede ser potencialmente asignados en el mismo periodo. La tercera matriz $C_{m,l}$, llamada matriz Salón-Características, da las características que cada salón posee donde el valor de una celda $c_{i,j}$ es 1 si

$i \in R$ tiene una característica $j \in F$; de lo contrario, el valor es 0. La cuarta matriz de $D_{n,l}$, llamada matriz de Evento-Características, da las características requerido por cada evento. Significa que caso $i \in E$ necesita característica $j \in F$ si y sólo si $d_{ij} = 1$. La última matriz $G_{n,m}$, llamada Evento-Salón, se enumeran los posibles salones a las que cada evento puede ser asignado. A través de esta matriz, podemos identificar rápidamente todos los salones que son adecuados en tamaño y función de cada caso. Por lo general, una matriz se utiliza para asignar a cada evento a un salón r_i y un periodo t_i . Cada par de (r_i, t_i) se le asigna un número particular que corresponde a un evento. Si un salón r_i en un periodo t_i es libre o ningún evento es colocado, a continuación, "-1" se asigna a ese par. De este modo, se asegura que no habrá más de un evento asignado al mismo par de modo que una de las restricciones duras será siempre satisfecha.

El algoritmo fue probado en casos de problemas discutidos en [6]. Ellos lidiaron con las siguientes restricciones duras:

RO'1. Ningún estudiante asiste a más de una clase al mismo tiempo.

RO'2. El salón es lo suficientemente grande para todos los estudiantes que asisten.

RO'3. El salón satisface todas las características requeridas para la clase.

RO'4. Solamente una clase se dicta en un salón y en un periodo.

También hay restricciones blandas, que son igualmente penalizadas por el número de sus violaciones y se describen como sigue:

RD'1. Un estudiante tiene una clase en el último periodo del día.

RD'2. Un estudiante tiene más de 2 clases consecutivamente.

RD'3. Un estudiante tiene una sola clase en un día

3.2 Descripción del Algoritmo

El algoritmo GSGA propuesto en [6] incorpora una estrategia de búsqueda guiada y un algoritmo de búsqueda local LS1 en el GA para resolver el UCTP. El pseudocódigo de GSGA para el UCTP se muestra en la Figura 3.1, en donde en primer lugar inicializamos la población al azar por la creación de cada individuo a través de la asignación de un intervalo de tiempo aleatorio para cada evento de acuerdo con una distribución uniforme y aplicando el algoritmo de coincidencia para asignar un salón

para el evento. Entonces, el algoritmo LS1 se aplica a cada miembro de la población inicial usando las tres estructuras vecinales N1, N2 y N3, que se describen como sigue:

N1: Operador que mueve un evento desde periodo a otro diferente.

N2: Operador que intercambia los periodos de dos eventos.

N3: Operador que permuta tres eventos en tres periodos diferentes en una de las dos posibles formas distintas.

Después de la inicialización de población, se construye una estructura de datos llamada MEM, que almacena una lista de pares de aulas y de intervalos de tiempo (r, t) para cada evento que tiene una penalidad cero (es decir, no ha habido violación de restricciones duras ni blandas) de los individuos seleccionados de la población. Después de eso, la estructura MEM se puede utilizar para guiar la generación de descendientes de las generaciones siguientes. La Estructura de datos MEM que se reconstruye periódicamente, por ejemplo cada τ generaciones.

En cada generación de GSGA, un descendiente se genera ya sea mediante el uso de MEM o aplicando el operador de cruce, en función de una probabilidad de γ . Después de eso, el descendiente será mutado por un operador de mutación con una probabilidad P_m seguido por la operación de búsqueda local LS1. Por último, el peor miembro en la población se sustituye por el descendiente recién generado. Esta iteración continúa hasta que una condición de término se alcanza, por ejemplo, un límite de tiempo preestablecido t_{\max} se alcanza.

Algorithm 3 Guided Search Genetic Algorithm (GSGA)

```

1: input : A problem instance I
2: randomly initialize a population of solutions
3: evaluate the individuals in the population
4: apply the local search scheme LS1 to each individual of
   the population
5: set the generation counter  $g := 0$ 
6: while the termination condition is not reached do
7:   if  $(g \bmod \tau) == 0$  then
8:     apply ConstructMEM() to construct MEM
9:   end if
10:  create a child using GuidedSolutionConstruction()
    or Crossover() with a probability  $\gamma$ 
11:  apply mutation to the child with a probability  $P_m$ 
12:  evaluate the child solution
13:  apply local search LS1 to the child
14:  replace the worst member of the population by the child
15:   $g := g + 1$ 
16: end while
17: output : The best solution  $s_{best}$  achieved for I

```

Figura 3.1 Algoritmo Genético con Búsqueda Guiada y Búsqueda Local [1].

A continuación se describen en detalle los componentes clave de GSGA.

3.3 Operador de Cruzamiento

La Figura 3.2 muestra el operador de cruce donde se seleccionan dos individuos de la población como los padres, por la selección torneo, con el tamaño del torneo igual a 2. Luego, para cada evento del descendiente, seleccionamos al azar uno de los padres y asignamos el periodo correspondiente para ese evento. Por último, asignamos los salones a los eventos en cada periodo no vacío utilizando el algoritmo de coincidencia.

Algorithm 6 *Crossover()*

```

1: input : The current population
2: Select parents  $P1$  and  $P2$  by the tournament selection
3: for each event  $e_i$  of the child  $Ch$  do
4:   if  $rand(0, 1) < 0.5$  then
5:      $e_i$  of  $Ch \leftarrow$  the time slot allocated to  $e_i$  of  $P1$ 
6:   else
7:      $e_i$  of  $Ch \leftarrow$  the time slot allocated to  $e_i$  of  $P2$ 
8:   end if
9: end for
10: allocate rooms to all occupied time slots using the match-
    ing algorithm
11: output : A new child generated using Crossover()

```

Figura 3.2 Pseudocódigo del operador de Cruzamiento [1].

3.4 Búsqueda Local LS1

El algoritmo LS1 comprueba las violaciones de restricciones duras de cada evento mientras hace caso omiso a las violaciones de restricciones blandas. Si hay violaciones de restricciones duras para un evento, trata de resolverlos mediante la aplicación de movimientos con las estructuras vecinales N1, N2 y N3. Después de cada movimiento, se aplica el algoritmo de coincidencia de los periodos afectados por el movimiento. Si no hay movimientos por probar se continúa con el próximo evento. Después de aplicar las estructuras vecinales en cada evento, si todavía hay alguna violación de restricción dura, entonces LS1 se detendrá, de lo contrario, LS1 realiza el segundo paso (líneas 14-27 en la Figura 3.3).

En el segundo paso, después de alcanzar una solución factible, LS1 trata a las restricciones blandas de una manera similar, como en el caso anterior. Cuando finaliza

LS1, obtenemos un individuo posiblemente mejorado. En la Figura 3.3 se muestra el pseudocódigo del Algoritmo de Búsqueda Local LS1.

Algorithm 2 Local Search Scheme 1 (LS1)

```

1: input : Individual I from the population
2: for each event  $e_i \in E$  do
3:   if event  $e_i$  is infeasible then
4:     if there is untried move left then
5:       calculate the moves: first N1, then N2 if N1 fails,
       and finally N3 if N1 and N2 fail
6:       apply the matching algorithm to the time slots
       affected by the move to allocate rooms for events
7:       delta evaluate the result of the move
8:       if moves reduce hard constraints violation then
9:         make the moves and go to line 4
10:      end if
11:    end if
12:  end if
13: end for
14: if no any hard constraints remain then
15:   for each event  $e_i \in E$  do
16:     if event  $e_i$  has soft constraint violation then
17:       if there is untried move left then
18:         calculate the moves: first N1, then N2 if N1 fails,
         and finally N3 if N1 and N2 fail
19:         apply the matching algorithm to the time slots
         affected by the move to allocate rooms for
         events
20:         delta evaluate the result of the move
21:         if moves reduce soft constraints violation then
22:           make the moves and go to line 17
23:         end if
24:       end if
25:     end if
26:   end for
27: end if
28: output : A possibly improved individual I

```

Figura 3.3 Algoritmo de Búsqueda Local LS1 [1].

3.5 Extensión del GSGA

Shengxiang Yang y Sadaf Naseem Jat [1], proponen una versión extendida de GSGA, denotado EGSGA, para el problema de horarios. En EGSGA, un nuevo algoritmo de búsqueda local denotado como LS2 es propuesta por los autores y se combina con el algoritmo de búsqueda LS1 para mejorar la calidad de la solución. El algoritmo LS2 se utiliza inmediatamente después de LS1 en soluciones aleatorias de la población inicial, así como después de la creación de un descendiente a través del operador de cruce o de la estructura de datos MEM y el operador de mutación. La idea básica del LS2 es elegir un periodo de alta penalidad que puede tener un gran número de eventos relacionados con violaciones de restricciones duras y blandas y tratar de reducir los valores de penalización de los eventos involucrados.

El algoritmo LS2 primero selecciona aleatoriamente un porcentaje preestablecido de periodos de T . A continuación, se calcula la penalidad de cada intervalo de tiempo seleccionado y elige el peor periodo w_t , que tiene el mayor valor penalización. Después de tomar el peor periodo, LS2 aplica la estructura local N1 para cada evento de w_t y comprueba el valor de penalización de cada evento, antes y después de aplicar los movimientos. Si todos los movimientos juntos en w_t reducen las violaciones de restricciones duras y/o blandas, entonces se aplican los movimientos, de lo contrario, no se aplican los movimientos.

En general, con el algoritmo LS2 fue propuesta para que mejore los individuos de la población y aumentar la calidad de un horario factible mediante la reducción del número de violaciones de restricciones. En la Figura 3.4 se muestra el pseudocódigo del Algoritmo de Búsqueda Local LS2.

Algorithm 7 Local Search Scheme 2 (LS2)

```

1: input : Individual  $I$  after LS1 is applied
2:  $S :=$  randomly select a preset percentage of time slots from the
   total time slots of  $T$ 
3: for each time slot  $t_i \in S$  do
4:   for each event  $j$  in time slot  $t_i$  do
5:     calculate the penalty value of event  $j$ 
6:   end for
7:   sum the total penalty value of events in time slot  $t_i$ 
8: end for
9: select the time slot  $w_t$  with the biggest penalty value from  $S$ 
10: for each event  $i$  in  $w_t$  do
11:   calculate a move of event  $i$  in the neighbourhood structure N1
12:   apply the matching algorithm to the time slots affected by the
      move
13:   compute the penalty of event  $i$  and delta evaluate the result
14: end for
15: if all the moves together reduce hard or soft constraint violations
      then
16:   apply the moves
17: else
18:   delete the moves
19: end if
20: output : A possibly improved individual  $I$ 

```

Figura 3.4 Algoritmo de Búsqueda Local LS2 [1].

Capítulo 4: Adaptación del Algoritmo Genético

El UCTP se afrontará tomando como caso de estudio particular a la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos.

Se propone un modelo matemático en el cual se definen las restricciones del problema, y luego se establece el diseño de la solución y la adaptación del algoritmo a esta. Para resolver el problema planteado se utiliza el EGSGA [1], cuyos resultados experimentales que se hicieron al compararlo con otros algoritmos (RIIA, NLGD, GAWLS, HEA, GBHH, THHS, LS, GA, AA, FA), en los tres tamaños de instancia (pequeño, mediano y grande) fue que el EGSGA es mejor en el 90.90% de los casos.

4.1 Problema

La definición del problema de estudio de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos se encuentra en la sección 1.2, este difiere a la definición del problema planteado por el autores Shengxiang Yang y Sadaf Naseem Jat [1] el cual se detalla en la sección 3.1 en los siguientes puntos:

D1. Las restricciones duras y blandas son particulares para cada institución, en nuestro caso contamos con 8 restricciones duras y 4 restricciones blandas, a diferencia del problema descrito por el autor que abarca solamente restricciones generales las cuales son 4 restricciones duras y 3 restricciones blandas, con lo que se han creado estructuras de datos adicionales para cubrir estas nuevas restricciones.

D2. La lista de los dictados de clases para asignar a un individuo de la población se encuentra ordenada según la diferencia entre la cantidad de horas disponibles del docente asignado y la cantidad de horas de carga lectiva.

D3. Se está considerando para los dictados de clase un mínimo de 2 periodos comparado con los dictados de clase del algoritmo a adaptar que solo tiene 1 periodo, con lo que hay diferencias en el cruzamiento, mutación y compactación porque se debe hacer los movimientos en bloques.

D4. Se está añadiendo un algoritmo de compactación en el algoritmo adaptado, ya que en algunos casos existen periodos de 1 o más entre dictados de clase donde no se puede asignar otros dictados de clase y esto conlleva a violar la restricción RO5.

D5. Se está considerando la disponibilidad de docentes para dictados de clase por lo que ha añadido la matriz Docente-Periodo para cumplir con la restricción RO5.

D6. Se está considerando la disponibilidad de los salones en la generación de los horarios debido a que es un recurso que puede estar ocupado en ciertos periodos y es necesario la matriz Salón-Periodo.

D7. Se está considerando la disponibilidad de en donde los dictados de clase se puede dictar ya que esto se debe a las asignaturas por ciclo académico se deben dictar en determinada franja de horario para esto utilizamos la matriz Dictado de Clase-Periodo para satisfacer la restricción RD7.

D8. Se está trabajando con dictados de clase fijados para los docentes de otras facultades y docentes que desean continuar con los horarios de ciclos anteriores, también se está trabajando con dictados de clase no fijados para los profesores que desean impartir sus clases dentro de su disponibilidad horaria.

D9. La matriz Evento-Estudiante descrita en el problema del autor nos indica que estudiante asiste a que evento, en el algoritmo adaptado no se está utilizando esta estructura ya que en nuestro caso se genera el horario y posteriormente los alumnos de matriculan.

D10. La estructura MEM no se ha implementado en el algoritmo adaptado.

4.2 Modelo General

4.2.1 Parámetros

Se definen los siguientes parámetros:

4.2.1.1 Índices

Se definen los siguientes índices que se consideran para el algoritmo:

m = Cantidad de dictados de clases.

p = Cantidad de docentes.

z = Cantidad de salones.

q = Cantidad de días de dictado.

y = Cantidad de periodos semanales.

It = Cantidad de iteraciones.

T_m = Tamaño de la población. $\forall m, p, z, q, y, It, T_m \in \mathbb{Z}$

P_c = Probabilidad de cruzamiento.

P_m = Probabilidad de mutación. $P_c \in [0,1], P_m \in [0,1]$

$MaxCons$ = Máximo de horas consecutivas de clases de un mismo ciclo en un día.

$MaxTotal$ = Máximo de horas de clases de un mismo ciclo en un día.

4.2.1.2 Se definen los siguientes vectores:

Sea $C = \{c_1, \dots, c_m\}$, el vector de dictado de clase.

Sea $D = \{d_1, \dots, d_p\}$, el vector de docentes que pueden dictar clases.

Sea $S = \{s_1, \dots, s_z\}$, el vector de salones donde se puede dictar una clase.

Sea $T = \{t_1, \dots, t_y\}$, el vector de periodos de tiempo en el que se puede dictar un curso cualquiera.

4.2.1.3 Se definen las siguientes matrices:

4.2.1.3.1 CC_{mm} la matriz Dictado de Clase - Conflictos

Indica si dos dictados de clase se pueden programar en el mismo intervalo de tiempo o no. Esto ayuda a identificar rápidamente las clases que pueden ser potencialmente asignados al mismo intervalo de tiempo.

$$CC_{mm'} = \begin{cases} 1, & \text{si las clases } c_m \text{ y } c_{m'} \text{ no se pueden dictar en el mismo periodo.} \\ 0, & \text{caso contrario.} \end{cases}$$

Por ejemplo: Los dictados de clases de un mismo ciclo de un mismo grupo no se pueden cruzar entre sí.

La matriz CC_{mm} se define como:

	C_1	C_2	...	C_m
C_1	0	0		0
C_2	1	0		1
...				
C_m	1	0		0

Tabla 4.1 Matriz Dictado de Clase – Conflictos [Fuente: Elaboración Propia].

4.2.1.3.2 DP_{py} la matriz Docente - Periodo

Muestra para cada docente los periodos disponibles en el que se puede asignar clases.

$$DP_{py} = \begin{cases} 1, & \text{si el docente } d_p \text{ se le puede asignar una clase en el periodo } t_y. \\ 0, & \text{caso contrario.} \end{cases}$$

La matriz DP_{py} se define como:

	T_1	T_2	...	T_y
D_1	0	0		0
D_2	1	1		1
...				
D_p	1	0		1

Tabla 4.2 Matriz Docente – Periodo [Fuente: Elaboración Propia].

4.2.1.3.3 CS_{mz} la matriz Dictado de Clase - Salón

Muestra para cada dictado de clase los salones que cuenta con las características necesarias para el dictado de la clase.

$$CS_{mz} = \begin{cases} 1, & \text{si el dictado de clase } c_m \text{ puede ser dictado en el salón } s_z. \\ 0, & \text{caso contrario.} \end{cases}$$

La matriz CS_{mz} se define como:

	S ₁	S ₂	...	S _z
C ₁	0	0		0
C ₂	1	1		1
...				
C _m	1	0		1

Tabla 4.3 Matriz Dictado de Clase – Salón [Fuente: Elaboración Propia].

4.2.1.3.4 CP_{my} la matriz Clase - Periodo

Muestra para cada dictado de clase los periodos válidos en el que se puede asignar un salón y un docente.

$$CP_{my} = \begin{cases} 1, & \text{si la clase } c_m \text{ puede ser dictado en el periodo } t_y. \\ 0, & \text{caso contrario.} \end{cases}$$

La matriz CP_{my} se define como:

	T ₁	T ₂	...	T _y
C ₁	0	0		0
C ₂	1	1		1
...				
C _m	1	0		1

Tabla 4.4 Matriz Dictado de Clase – Periodo [Fuente: Elaboración Propia].

4.2.1.3.5 SP_{zy} la matriz Salón - Periodo

Muestra las clases junto con el docente asignado en un periodo para cada salón.

$$SP_{zy} = \begin{cases} c_m d_p, & \text{si el salón } s_z \text{ tiene asignado una clase y un docente en el periodo } t_y. \\ \text{nulo}, & \text{caso contrario.} \end{cases}$$

La matriz SP_{zy} se define como:

	T ₁₁	T ₂₁	...	T _{yq}
S ₁	C ₁₀ , D ₃	C ₁₀ , D ₃		C ₃₅ , D ₂
S ₂	C ₁₀ , D ₃	C ₁₀ , D ₃		C ₁₀ , D ₃
...				
S _z	C ₁₀ , D ₃	C ₁₀ , D ₃		C ₁₀ , D ₃

Tabla 4.5 Matriz Salón – Periodo [Fuente: Elaboración Propia].

4.2.2 Variables

4.2.2.1 Variables de decisión

$$X_{ijkl} \begin{cases} \text{El salón } i \text{ es asignado a la clase } j \text{ con el docente } k \text{ en el periodo } l. \\ 0 \text{ si no.} \end{cases}$$

$$\forall i \in \{1, \dots, z\}, \forall j \in \{1, \dots, m\}, \forall k \in \{1, \dots, p\}, \forall l \in \{1, \dots, y\}$$

4.2.3 Restricciones

4.2.3.1 Restricciones Obligatorias:

R.O.1. Una clase debe tener asignado a lo más un docente en un salón en un periodo específico.

$$\sum_{i=1}^z \sum_{k=1}^p X_{ijkl} \leq 1 \quad \forall j \forall l$$

En un periodo se debe asignar una clase y un docente en sólo un salón (valida que la misma clase solo se asigne una vez en un periodo), por ello recorreremos las filas de los salones en un periodo y se debe de tener asignado solo 1 vez una clase y docente fijo.

R.O.2. Un docente debe tener asignado a lo más una clase en un salón en un periodo específico.

$$\sum_{i=1}^z \sum_{j=1}^m x_{ijkl} \leq 1 \quad \forall k \forall l$$

Esta restricción nos permitirá validar que no se asignen más de una vez un docente en un mismo periodo, para ello se recorrerán todos los salones y se validará para cada docente que en un periodo este asignado como máximo una sola vez.

R.O.3. Un salón puede tener a lo más una clase asignado y un docente en un periodo específico.

La estructura de la representación del problema, la matriz Salón - Periodo, solo puede contener una sola clase y un docente, por ende se cumple esta restricción.

R.O.4. Un docente tendrá asignado una lista de clases el cual dictará en el ciclo.

Se está considerando que a cada docente se está asignando las clases que dictará en el ciclo, por ello hay que validar que previo a iniciar al algoritmo se tenga asignado un docente a todas las clases en la matriz Docente - Clase.

R.O.5. No se puede asignar una clase a un docente fuera de su disponibilidad horaria.

$$Si \sum_{i=1}^z \sum_{j=1}^m x_{ijkl} \geq 1 \Leftrightarrow DP_{kl} = 1 \quad \forall k \forall l$$

Si se asigna una clase a un docente se debe verificar en la matriz Docente Periodo que el docente tenga la disponibilidad en ese periodo.

R.O.6. La capacidad del salón debe de ser mayor o igual a la cantidad de alumnos permitidos dentro de una clase.

$$Si \sum_{k=1}^p \sum_{l=1}^y x_{ijkl} \geq 1 \Leftrightarrow CS_{j,i} = 1 \quad \forall j \forall i$$

Si se asigna una clase a un salón en un periodo se debe verificar en la matriz Clase Salón que el salón sea apto para el dictado de la clase, es decir, el salón es de mayor o igual capacidad que requiere la clase.

R.D.7. Los dictados de clase perteneciente a un mismo grupo de cada ciclo académico no deben de solaparse entre ellos, ni en duración ni ubicación física, una vez asignada la hora de inicio de éstos y el salón correspondiente.

Si se asigna una clase a un salón en un periodo, se debe verificar para ese periodo y para todos los salones que las clases que tengan asignadas no tengan conflicto, ya que si el valor es 1 indica que esa clase no puede dictarse en el mismo periodo que la otra clase, para ello se tiene que validar:

$$Si \sum_{i=1}^z \sum_{k=1}^p x_{ijkl} \geq 1 \Leftrightarrow \sum_{j'=1}^m CC_{jj'} = 0 \quad \forall j \forall j' \forall l$$

j' : Son todos los cursos que están en el mismo periodo.

R.O.8. Las clases que son de teoría o de práctica o de laboratorio deben asignarse a salones que cumplan con las características necesarias para el dictado de la clase.

$$Si \sum_{k=1}^p \sum_{l=1}^y x_{ijkl} \geq 1 \Leftrightarrow CS_{j,i} = 1 \quad \forall j \forall i$$

Si se asigna una clase a un salón se debe verificar en la matriz Clase Salón que el salón sea apto para el dictado de la clase, es decir, en el salón se puede dictar el tipo de dictado de la clase.

4.2.3.2 Restricciones Deseables:

R.D.1. Las clases deberían dictarse dentro de una franja horaria, es decir, se debe asignar un intervalo de tiempo por día en donde la clase se pueda dictar de acuerdo al ciclo académico al cual pertenece.

Se tiene la matriz Ciclo – Periodo, aquí se tendrá definido para cada clase todos los periodos en el que se desea asignar y su valor debe ser 1.

R.D.2. Para las clases que pertenecen a un ciclo se debe limitar la cantidad de horas de dictado asignados en un mismo día con un máximo de 6 periodos consecutivos.

R.D.3. Para las clases que pertenecen a un mismo ciclo académico se debe limitar la cantidad de horas de dictado asignados en un mismo día con un máximo de 9 periodos.

R.D.4. Los intervalos de tiempo entre dictados de clases que pertenecen a un grupo de un mismo ciclo deben minimizarse.

4.3 Modelo Particular

Una vez descrito el modelo general, se realizará la adaptación para el problema de la Facultad de Ingeniería de Sistemas e Informática. La diferencia con el modelo general, será en la definición de los valores de los parámetros:

4.3.1 Parámetros

La cantidad de dictados de clases se define como la cantidad de asignaturas (55 asignaturas del plan de estudios 2009 y 18 asignaturas del plan de estudios 2014), la cantidad de modalidad de dictado que tiene cada asignatura y por la cantidad de grupos o secciones que se dictarán para un ciclo, entonces habría un total de 380.

La cantidad de docentes es de 125, cada uno pertenece a una categoría y se cuenta con su disponibilidad horaria.

La cantidad de salones es igual a 27.

La cantidad de periodos semanales se determinan en 84 periodos de tiempo posibles que van desde el lunes a sábado desde las 8 am hasta las 10:00 pm definiendo la duración por periodo de 60 minutos:

Periodo 1 Lunes 8am-9am	Periodo 2 Lunes 9am-10am	...	Periodo 84 Sábado 9pm-10pm
----------------------------	-----------------------------	-----	-------------------------------

Figura 4.1 Periodos de Clase [Fuente: Elaboración Propia].

La cantidad de días para dictado son 6: de lunes a sábado.

La cantidad de iteraciones será de 900, el tamaño de la población es de 50 individuos, la probabilidad de cruzamiento es de 0.6, la probabilidad de mutación es de 0.2, el número máximo de horas consecutivas de clases de un mismo ciclo en un día es de 6 horas y el número máximo de horas de clases de un mismo ciclo en un día es de 9 horas.

4.3.2 Variables

4.3.2.1 Variables de decisión

$$X_{ijkl} = \begin{cases} \text{El salón } i \text{ es asignado a la clase } j \text{ con el docente } k \text{ en el periodo } l. \\ 0 \text{ si no.} \end{cases}$$

$$\forall i \in \{1, \dots, 27\}, \forall j \in \{1, \dots, 336\}, \forall k \in \{1, \dots, 103\}, \forall l \in \{1, \dots, 84\}$$

4.4 Función Objetivo

La meta es minimizar el número de restricciones blandas de un horario factible (una solución factible es aquella en donde no existe violaciones de restricciones duras). La función objetivo $f(s)$ para un horario s es:

$$f(s) = \#hcv(s) * C + \#scv(s)$$

Donde:

$\#hcv$ = número de violaciones de restricciones duras.

$\#scv$ = número de violaciones de restricciones blandas.

$C = 30$, es el valor que dará más importancia al número de violaciones de restricciones duras respecto a las restricciones blandas.

4.5 Representación del Problema

En la Figura 4.6 se muestra la representación del problema en donde las columnas representan los periodos y las filas representan los salones en la cuales se van a ir asignando los dictados de clase y los docentes asignados a estos.

	Periodo					
Salón	T1	T2	T3	T4	...	T84
101	(C ₁₉ , D ₃₂)	(C ₁₉ , D ₃₂)	(C ₄₃ , D ₂₁)	(C ₄₃ , D ₂₁)		
102	(C ₅ , D ₂₃)	(C ₅ , D ₂₃)	(C ₅ , D ₂₃)			
103	(C ₂ , D ₇₆)	(C ₂ , D ₇₆)				
104						
105						
106						
107						
108						
201						
202						
203						
204						
.						
.						
.						
Lab08						
Lab09						

Tabla 4.6. Matriz de Representación del Problema [Fuente: Elaboración Propia].

Con respecto al gen, será definido por el vector que se muestra en la Figura 4.2.

Dictado de Clase	Docente
------------------	---------

Figura 4.2. Representación del gen [Fuente: Elaboración Propia].

Las variables dictado de clase y docente tomarán sus valores correspondientes, es decir la variable dictado de clase tomará un valor entre 1 al 380, y la variable Docentes tomará un valor entre 1 al 125.

4.6 Algoritmo Adaptado

En la Figura 4.3, mostramos el algoritmo adaptado para el caso de la FISl, en donde en primer lugar inicializamos la población de tamaño 50, en donde asignamos los dictados de clases aleatoriamente, de acuerdo a lo descrito en D2, luego aplicamos el algoritmo de búsqueda local LS1 y LS2 para mejorar la calidad de los individuos, seleccionamos 2 padres aleatoriamente por la selección del torneo con $k = 2$, realizamos el operador de cruzamiento de los padres con una probabilidad de 0.7, en caso no se realice el cruzamiento seleccionamos uno de los padres, seguidamente realizamos el operador de mutación con una probabilidad de 0,3, luego aplicamos el algoritmo de búsqueda local LS1 y LS2, este proceso se repite hasta que la condición de termino se halla alcanzado, en nuestro caso la cantidad de iteraciones = 900;

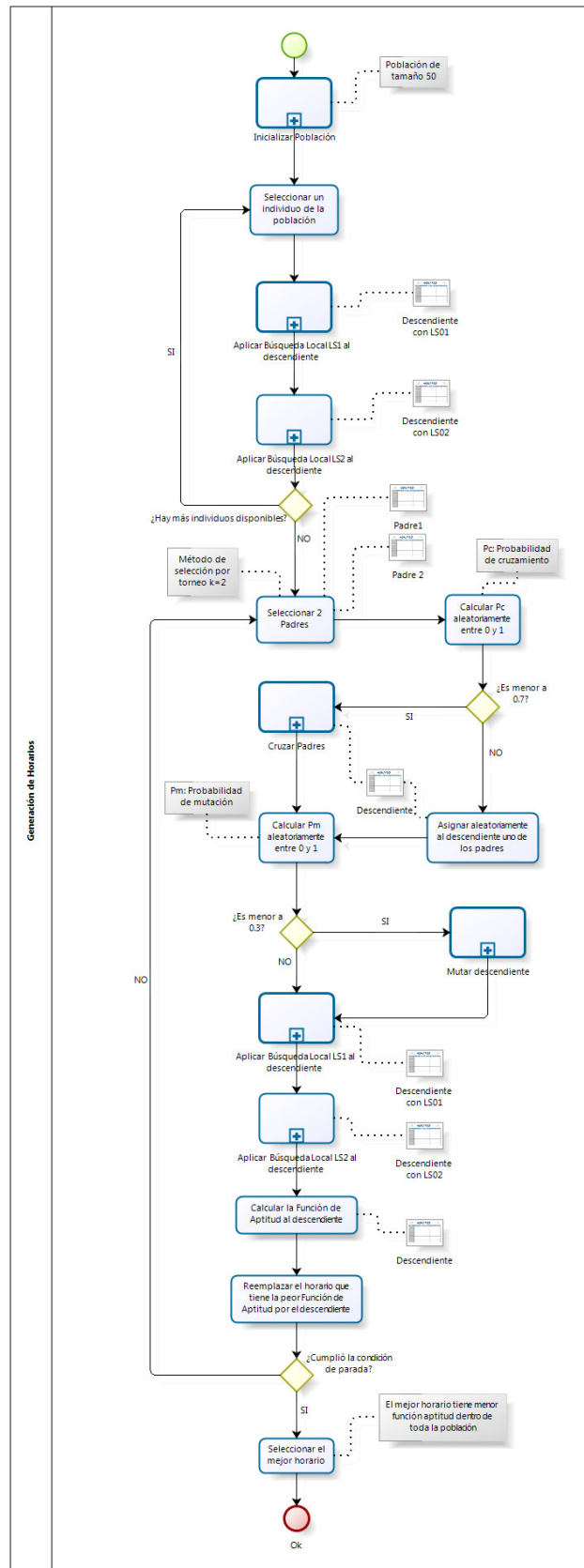


Figura 4.3 Algoritmo Genético Adaptado [Fuente: Elaboración Propia].

4.6.1 Descripción del Algoritmo

4.6.1.1 Inicialización de la Población

Se genera aleatoriamente la población inicial, que está constituida por un conjunto de individuos (horarios), los cuales representan las posibles soluciones del problema, a través de la asignación de un periodo aleatorio para cada clase de acuerdo a lo descrito en D2.

En la Figura 4.4 podemos observar el algoritmo de inicialización de la población en donde en una primera parte para cada individuo de la población asignamos los dictados de clase fijados en el periodo fijado previamente y en un salón aleatorio que sea válido (la capacidad del salón debe ser mayor o igual a la cantidad de alumnos estimado para el dictado de la clase y debe permitir la modalidad de dictado de dicha clase, además que el salón no se encuentre ocupado por un dictado de clase previamente asignada y que tenga disponibilidad en ese periodo); después de asignado los dictados de clase fijados, se debe asignar los dictados de clases no fijados en un periodo aleatorio y en un salón aleatorio válido.

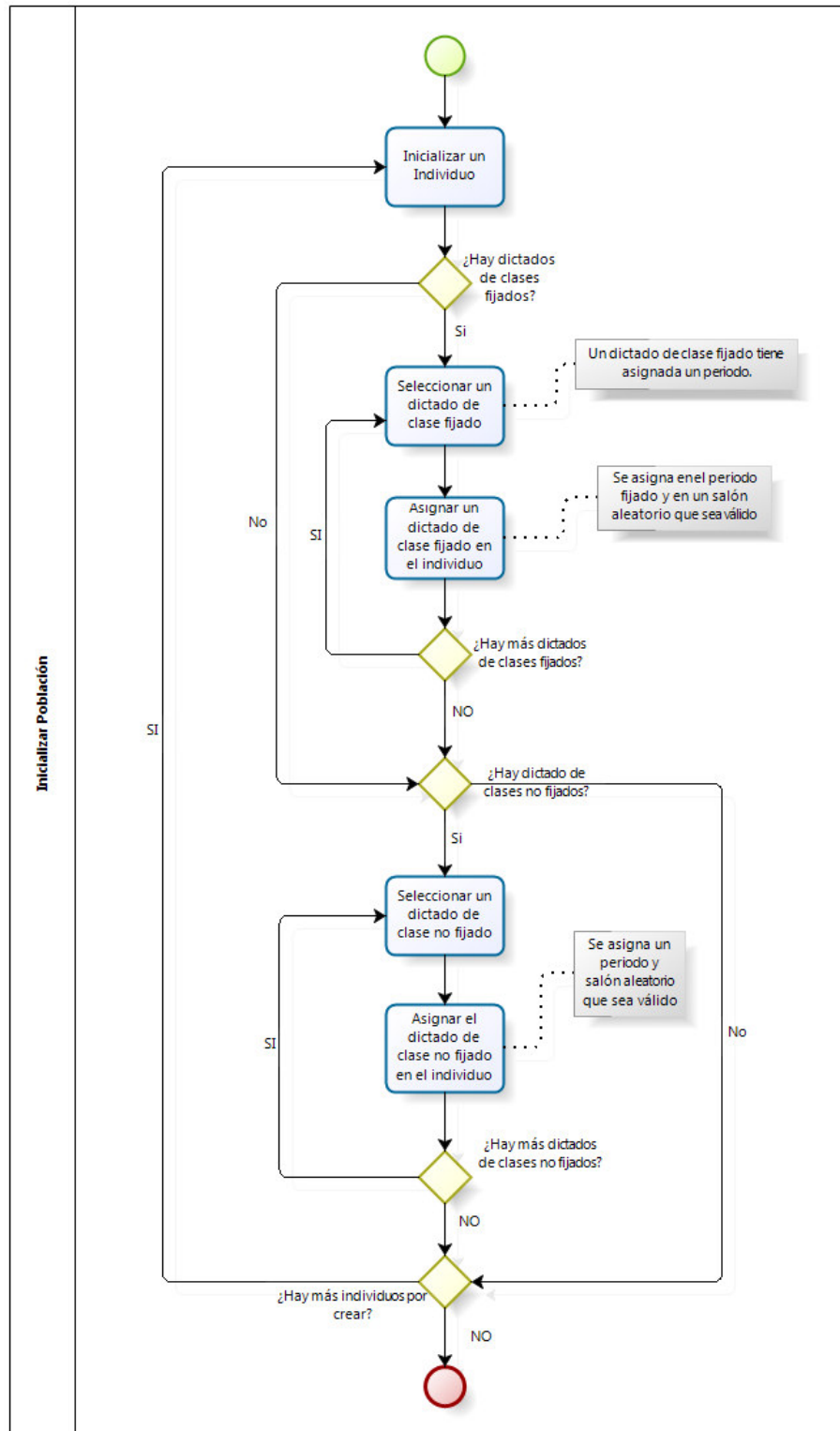


Figura 4.4 Algoritmo de Inicialización de la Población [Fuente: Elaboración Propia].

4.6.1.2 Evaluación de individuo

A cada uno de los individuos de esta población se aplicará la función de aptitud para saber qué tan buena es la solución que se está codificando.

4.6.1.3 Selección

La selección por torneo determinista con tamaño igual a 2 es la técnica utilizada para seleccionar los padres que serán cruzados en la siguiente generación. La cual consiste en seleccionar aleatoriamente un número p de individuos, en nuestro caso es 2. Después de entre los individuos seleccionados se elige el mejor, es decir, el de menor valor de adaptación (función de aptitud). El proceso se repite hasta completar el número de individuos que se desea seleccionar.

4.6.1.4 Recombinación o Cruzamiento

La recombinación es el principal operador genético, opera sobre dos individuos a la vez para generar un descendiente, se combinan las características de ambos individuos padres. En nuestro caso, en la selección ya se obtiene 2 padres y se realiza los siguientes pasos:

- Se inicializa un descendiente (horario vacío).
- Para cada dictado de clase fijado:

Paso 1. Se escoge uno de los 2 padres al azar.

Paso 2. Se busca el salón y el periodo del dictado de clase seleccionado en el padre seleccionado para colocarlo en el descendiente.

Paso 3. En caso se encuentre ya asignado otro dictado de clase en el descendiente, se buscará otro salón válido en el mismo periodo.

- Para cada dictado de clase no fijado:

Paso 1. Se escoge uno de los 2 padres al azar.

Paso 2. Se busca el salón y el periodo de la clase seleccionada en el padre seleccionado para colocarlo en el descendiente y debe cumplir que el salón

sea válido, la unicidad de docente y la disponibilidad de docente (restricciones 2 y 5).

Paso 3. En caso se encuentre ya asignado otro dictado de clase, se buscará otro salón válido en el mismo periodo y que no violen las restricciones 2 y 5.

Paso 4. En caso no se encuentren salones válidos se seleccione el otro Horario Padre y repetir los pasos 2 y 3.

Paso 5. En caso tampoco no se encuentren salones válidos con el segundo Horario Padre escogido seleccionamos la lista de todos los salones válidos para el dictado de clase seleccionado y lo colocamos en un periodo aleatorio que cumpla con las restricciones mencionadas en el paso 2.

Paso 6. En caso no es no se pueda asignar la clase a un salón y en un periodo que cumplan las restricciones mencionadas en el paso 2 se asignará en un periodo aleatorio y un salón valido.

Paso 7. Después de asignado el dictado de clase se realiza el algoritmo de compactación.

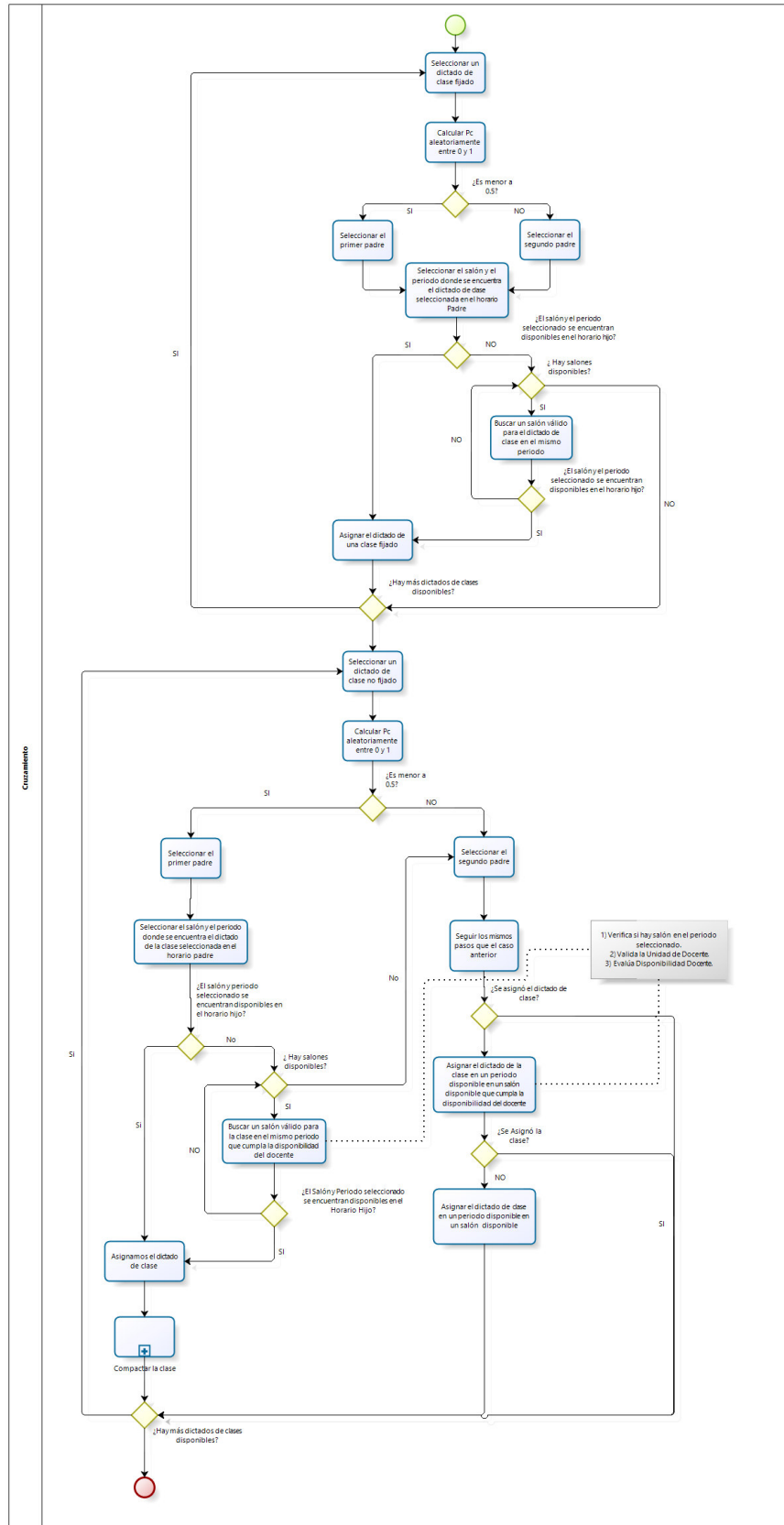


Figura 4.5 Cruzamiento [Fuente: Elaboración Propia].

4.6.1.5 Compactación

El algoritmo de compactación de dictados de clases mostrado en la Figura 5.6 se creó para eliminar los espacios en blanco (periodos sin dictados de clases asignados) en donde no se podían asignar dictados de clases y se realizan los siguientes pasos:

- Se escoge uno de los algoritmos de compactación hacia abajo o hacia arriba.
- Algoritmo de Compactación hacia abajo:

Paso 1. Seleccionamos el periodo 1 menor al periodo actual en el mismo salón y debe cumplir que el salón sea válido, la unicidad de docente y la disponibilidad de docente (restricciones 2, 5).

Paso 2. Si es que el periodo cumple las restricciones actualizamos el periodo actual por el nuevo periodo seleccionado y repetimos los pasos 1 y 2 hasta encontrar un periodo que no cumpla las restricciones 2 y 5.

Paso 3. Si es que no se logró compactar utilizamos el Algoritmo de Compactación hacia arriba.

- Algoritmo de Compactación hacia arriba:

Paso 1. Seleccionamos el periodo 1 periodo mayor al periodo actual en el mismo salón y debe cumplir que el salón sea válido, la unicidad de docente y la disponibilidad de docente (restricciones 2 y 5).

Paso 2. Si es que el periodo cumple las restricciones actualizamos el periodo actual por el nuevo periodo encontrado y repetimos los pasos 1 y 2 hasta encontrar un periodo que no cumpla las restricciones 2 y 5.

Paso 3. Si es que no se logró compactar utilizamos el Algoritmo de Compactación hacia abajo.

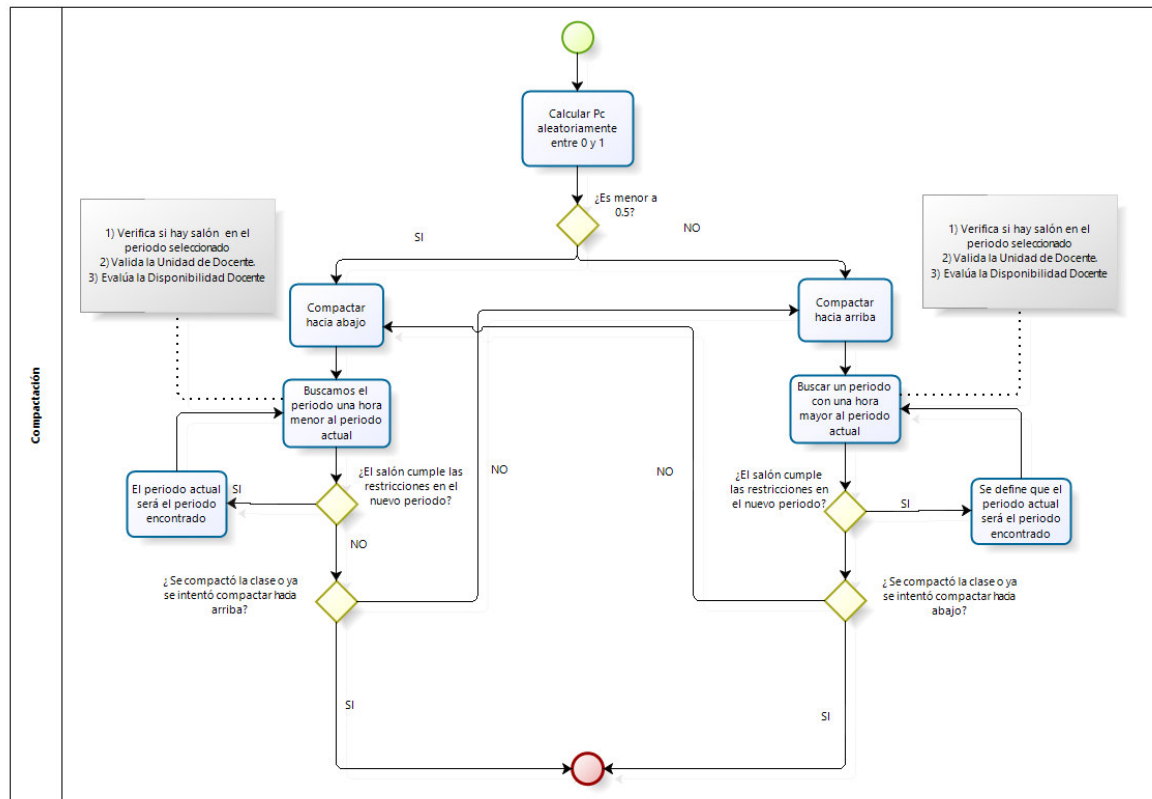


Figura 4.6 Algoritmo de Compactación [Fuente: Elaboración Propia].

4.6.1.6 Mutación

El operador de mutación consiste en aplicar N1 o N2 y eso se realiza con una probabilidad de P_m como se muestra en la Figura 5.7.

- Seleccionamos al azar uno de los 2 algoritmos de búsqueda local N1 o N2.
- N1:

Paso 1. Cambia 1 un dictado de clase no fijado a otro periodo diferente que esté vacío y cumpla con las restricciones duras, debe cumplir que el salón sea válido, la unicidad de docente, la disponibilidad de docente y la de conflictos de dictados de clases (restricciones 2, 5 y 7).

Paso 2. Una vez encontrado el periodo se procede a compactar el dictado de clase.

- N2:

Paso 4. Intercambia 2 dictados de clases no fijados que tenga la misma cantidad de horas de dictado.

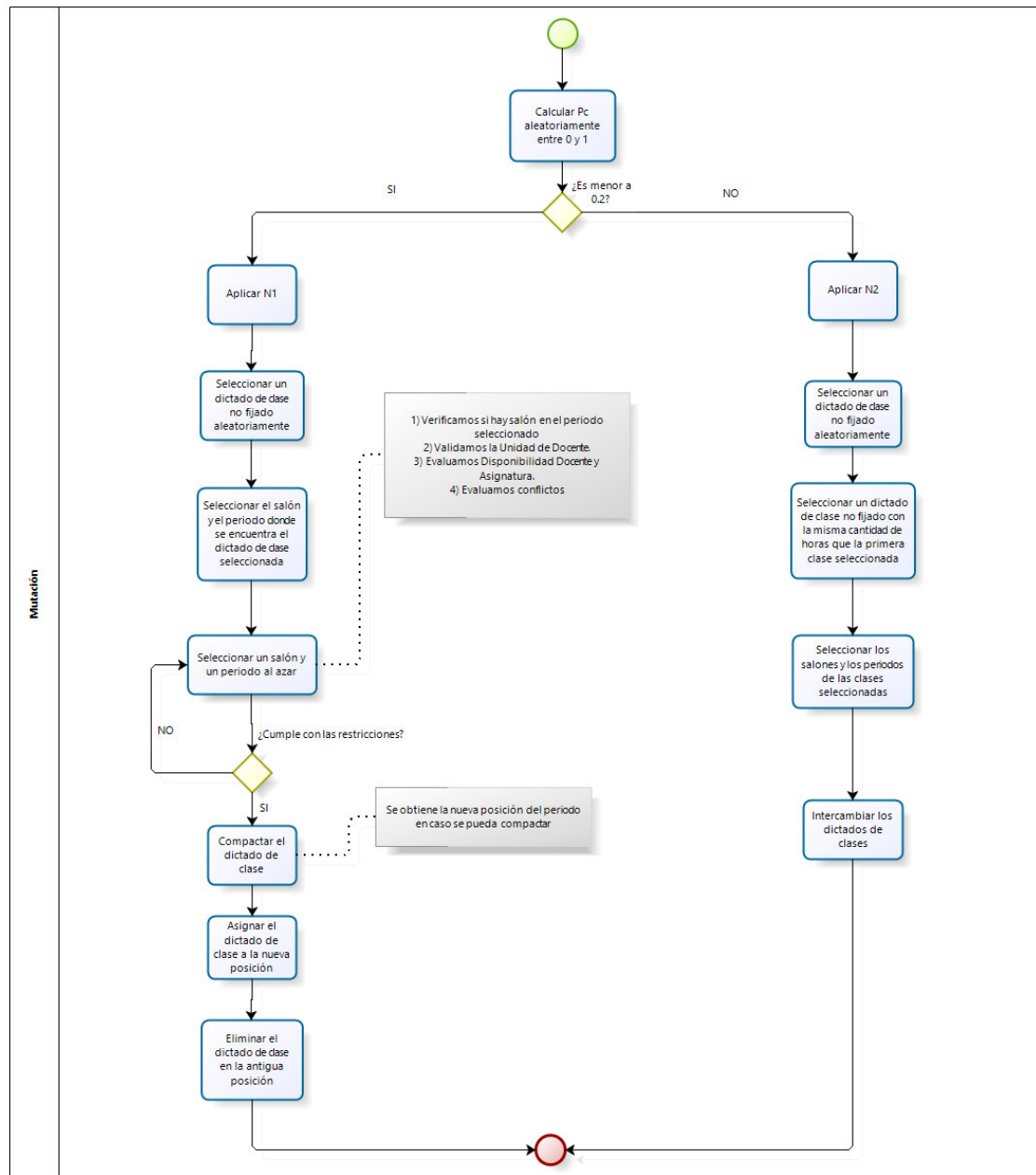


Figura 4.7 Algoritmo de Mutación [Fuente: Elaboración Propia].

4.6.1.7 LS1

Se aplica una búsqueda local a cada dictado de clase, es decir, para todos los dictados de clases se realiza el algoritmo N1 y se evalúa la función de aptitud del individuo para comparar si disminuyó o mantuvo el valor de la función de aptitud, en caso contrario se aplicará el algoritmo N2 y se evalúa la función de aptitud del individuo para comparar si disminuyó o mantuvo la función de aptitud en cuyo caso se aplican los cambios como se

describe en la Figura 4.8.

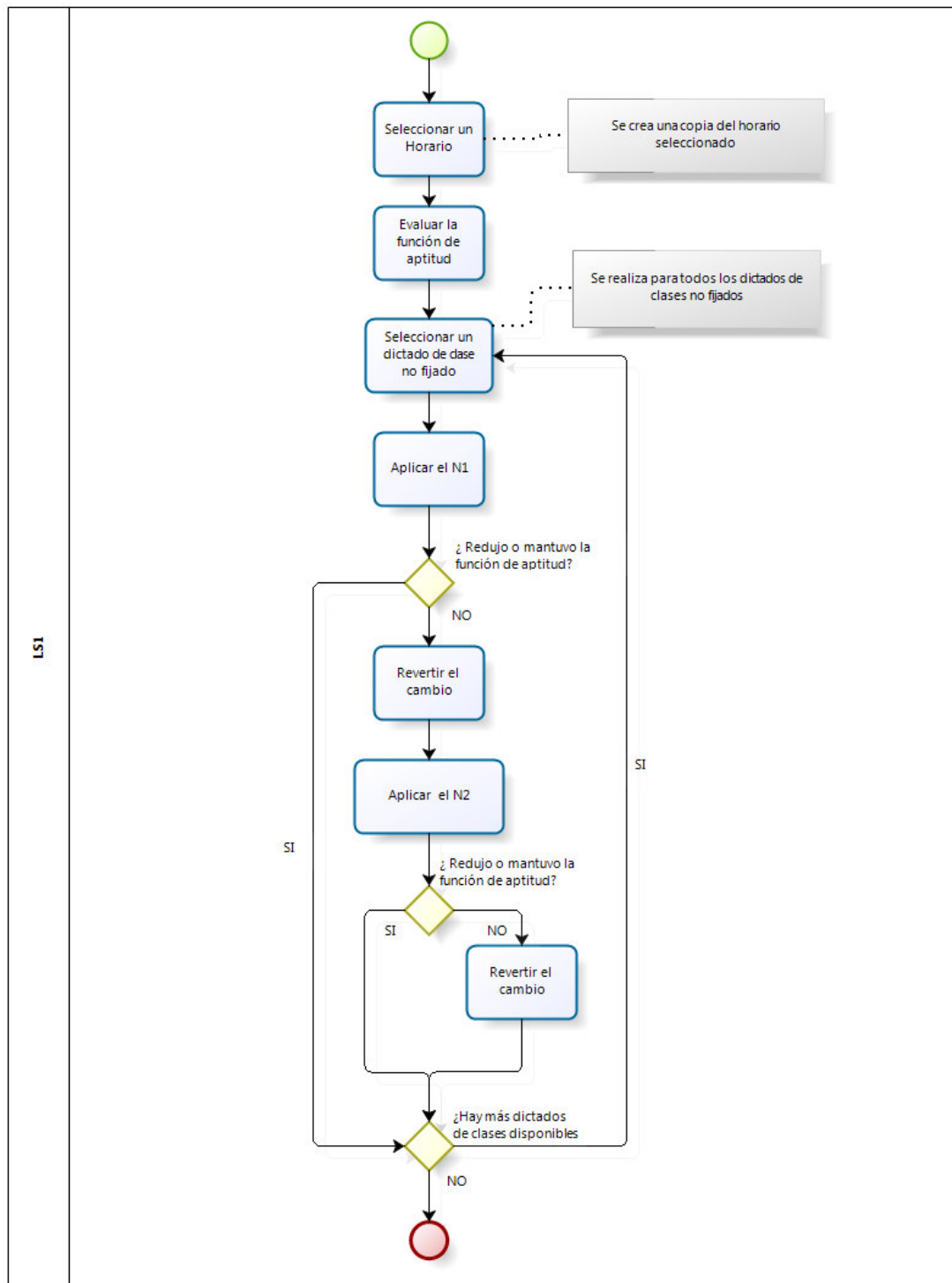


Figura 4.8 Algoritmo LS1 [Fuente: Elaboración Propia].

4.6.1.8 LS2

El algoritmo de búsqueda local LS2 que se muestra en la Figura 4.9 busca reducir la función de aptitud obteniendo el periodo con más conflictos, una vez obtenido el periodo, para todos los dictados de clases que pertenecen a ese periodo aplicamos el algoritmo de búsqueda local N1, si es que todos los movimientos juntos reducen o mantienen la función de aptitud se aplican los movimientos.

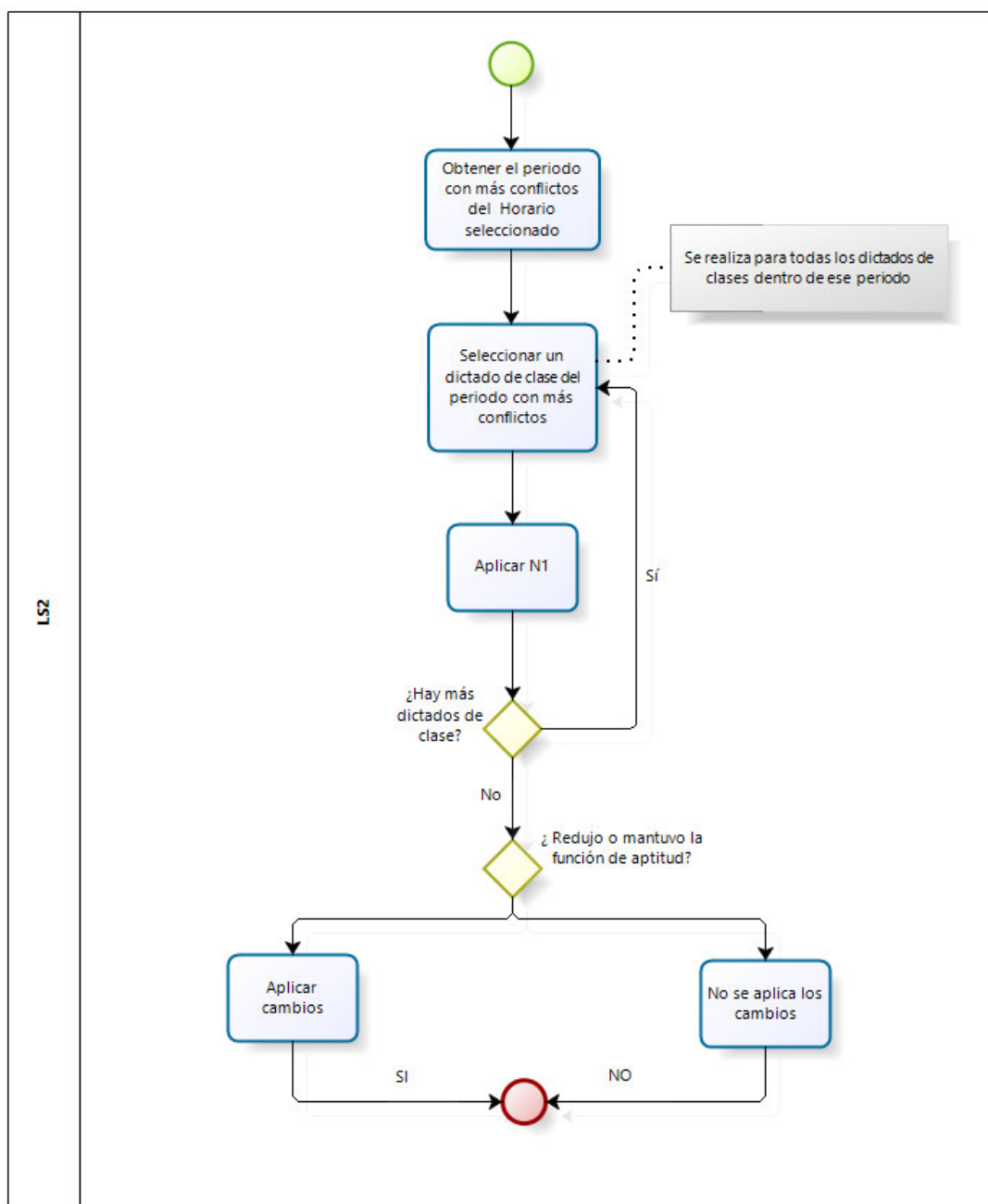


Figura 4.9 Algoritmo LS2 [Fuente: Elaboración Propia].

4.6.1.9 Reemplazo

Una vez aplicados los operadores genéticos, se selecciona el peor individuo de la población, es decir, el individuo que tenga el mayor valor de la función de aptitud y se reemplaza por el descendiente creado.

4.6.1.10 Condición de término

El AG se deberá detener cuando se alcance la solución óptima, pero ésta generalmente se desconoce, por lo que se deben utilizar otros criterios de detención. Normalmente se usan dos criterios: correr el AG un número máximo de iteraciones (generaciones) o un máximo número de tiempo. En nuestro caso, la condición de término es la cantidad de iteraciones.

Capítulo 5: Sistema de Generación de Horarios

En este capítulo se dará a conocer la metodología de desarrollo del software utilizado para la solución del problema tratado en este trabajo. Se mostrarán los requerimientos mínimos de hardware y software para su correcto funcionamiento, luego se detallará el modelo de la base de datos y por último los módulos funcionales del sistema.

5.1 Definición de la Metodología de a usar

La metodología a usar es el R.U.P debido a que puede especializarse a una gran variedad de sistemas software para diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes tamaños de proyectos.

La figura 5.1 muestra en la columna izquierda los flujos de trabajo: requisitos, análisis, diseño, implementación y prueba. Las curvas son una aproximación de hasta dónde se llevan a cabo los flujos de trabajo de cada fase, en donde cada fase se divide normalmente en iteraciones o mini-proyectos.

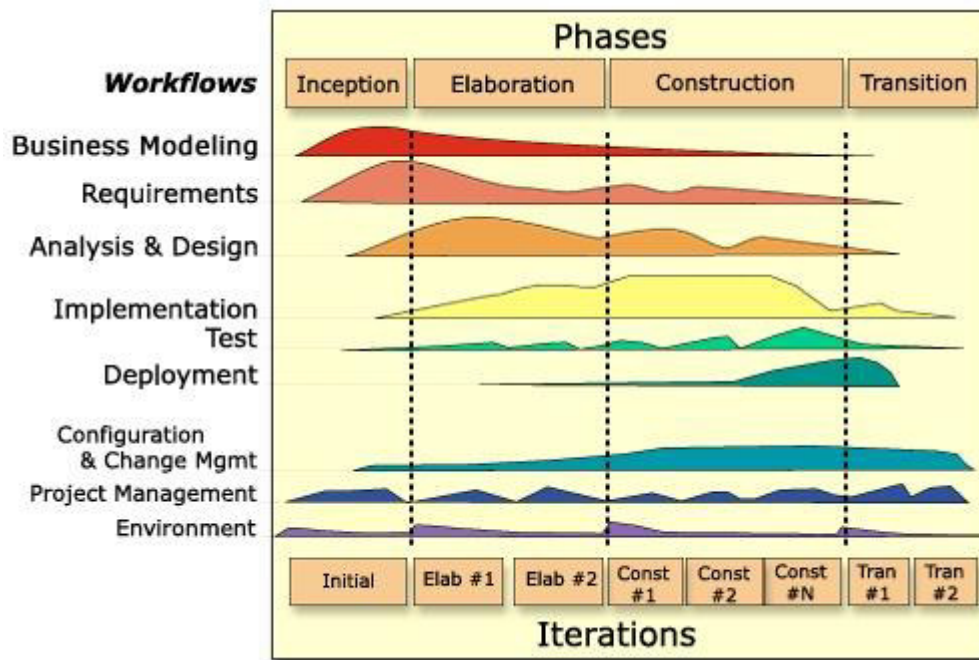


Figura 5.1 Los cinco flujos de trabajo sobre las cuatro fases: inicio, elaboración, construcción y prueba [46].

5.2 Requerimientos:

Esta disciplina toma el modelo de objetos de negocio y el modelo de casos de uso del negocio como entrada principal y trata de entender el problema que se quiere resolver por medio del sistema que se va a desarrollar.

En el siguiente paso se definen las necesidades de los clientes las cuales se plasman en los diagramas de casos de uso que capturan los requerimientos funcionales de este modo los stakeholders y los desarrolladores logran un entendimiento común de lo que se quiere realizar. La especificación de los casos de uso más importantes se verán en el Anexo 1.

5.2.1 Requisitos Funcionales

El sistema de Generación de Horarios se ha dividido en 5 subsistemas como se muestra en la Figura 5.3:

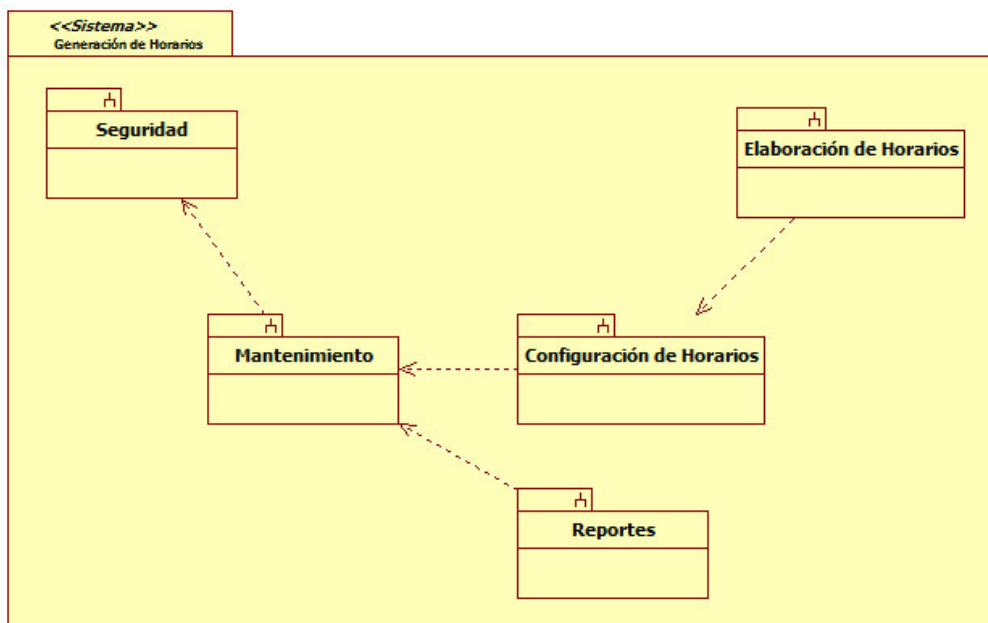


Figura 5.2 Diagrama de paquetes [Fuente: Elaboración Propia].

5.2.2 Perfiles de Usuarios:

En la Tabla 5.1 se muestra la lista de perfiles del sistema con la descripción de la responsabilidad de cada perfil en el sistema.

Perfil	Descripción
Jefe de Departamento	Responsable de realizar el mantenimiento a los docentes y de la asignación de la carga académica en cada ciclo académico.
Director de Escuela	Es el encargado de mantener los datos generales que se requieren como plan de estudios, las asignaturas, las clases y la cantidad de grupos que se abrirán para un determinado ciclo que debe registrar.
Docente	Es el responsable de actualizar su disponibilidad horaria para un nuevo ciclo, además de las clases que le gustaría dictar.
Asistente de Logística	Responsable de realizar la actualización de los datos de los salones: la creación y eliminación de salones, la modificación de la capacidad con la que cuentan y la disponibilidad de la misma.
Asistente de Laboratorio	Responsable de realizar la actualización de los datos de los laboratorios: la creación y eliminación de salones, la modificación de la capacidad con la que cuentan y la disponibilidad de la misma.
Administrador	Tiene todas las funcionalidades de los otros roles, además sus propias tareas.

Tabla 5.1 Lista de Perfiles [Fuente: Elaboración Propia].

La generalización entre los perfiles se muestra en la Figura 5.3:

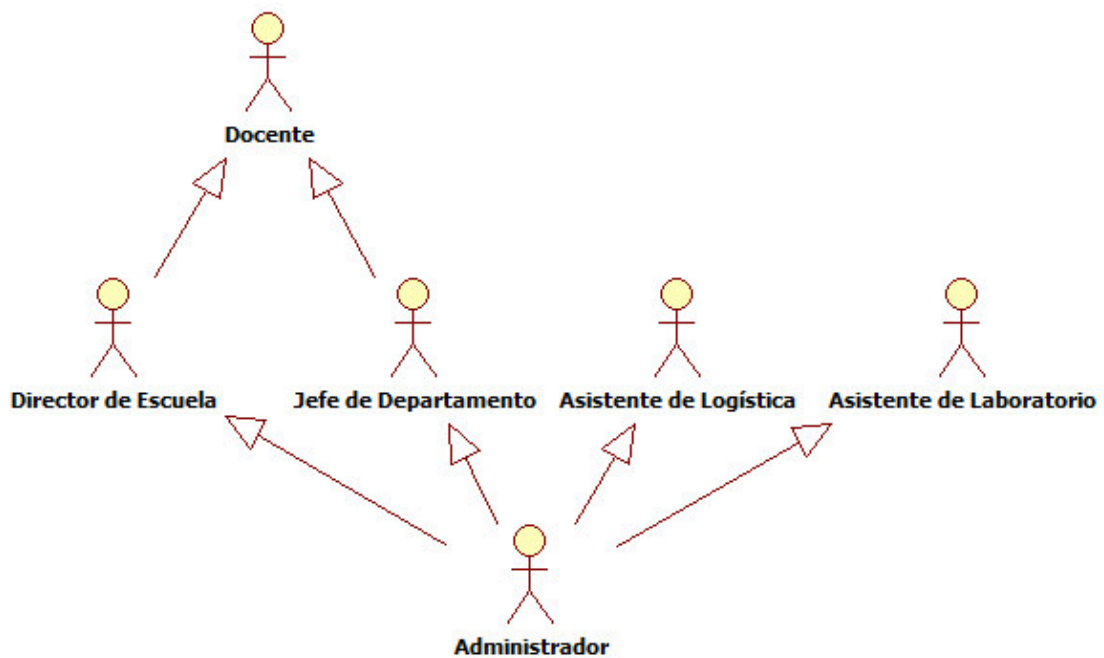
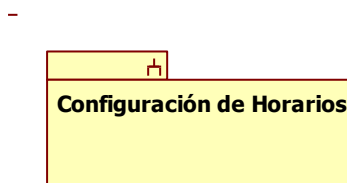


Figura 5.3 Diagrama de perfiles de usuarios [Fuente: Elaboración Propia].

5.2.3 Principales Subsistemas

5.2.3.1 Módulo de Configuración de Horarios

A través de este subsistema se podrá configurar los parámetros del algoritmo, la disponibilidad horaria de los profesores, así como la carga académica de los docentes como se muestra en el diagrama de casos de usos, la Figura 5.4.



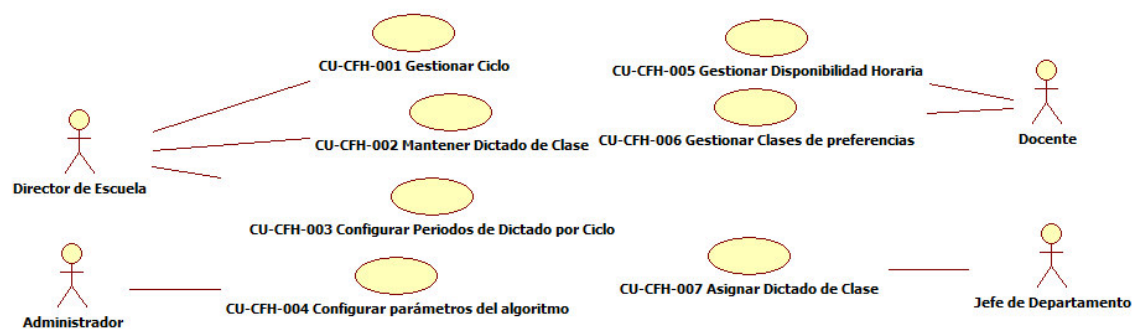


Figura 5.4 Diagrama de Casos de Uso del Módulo Configuración de Horarios [Fuente: Elaboración Propia].

El propósito u objetivo de cada caso de uso y la importancia por su impacto en la arquitectura del software se presenta a continuación:

5.2.3.1.1 CU-CFH-001: Gestionar Ciclo

Este caso de uso es realizado por el Director de Escuela para habilitar un nuevo ciclo para una Dirección de Escuela en el que se deberá generar horarios como se muestra en la Figura 5.5.

Nombre	Fecha Inicial	Fecha Final	Estado
2015-II	01-04-2015	31-07-2015	Terminado
2015-I	01-01-2015	31-03-2015	Activo
2015-0		11-12-2014	Terminado
2014-I		08-04-2014	Terminado
2014-0		08-08-2013	Terminado
2013-II		08-08-2013	Terminado
2013-I		08-08-2013	Terminado
2013-0		08-08-2013	Terminado
2012-II		08-08-2013	Terminado
2012-I		08-08-2013	Terminado

Figura 5.5 Gestión de Ciclo [Fuente: Elaboración Propia].

5.2.3.1.2 CU- CFH-002: Mantener Dictado de Clase

Este caso de uso es realizado por el Director de Escuela quien es responsable de registrar dictados de clases que se abrirán para un ciclo vigente como se muestra en la Figura 5.6.

Lista de Secciones - Ciclo : 2015-I

Registrar Seleccione el Ciclo: 2015-I Seleccione el Ciclo Académico: Primer Ciclo

Calculo I

Grupo	Modalidad
Grupo1	Teórico
Grupo1	Práctico
Grupo2	Teórico
Grupo2	Práctico
Grupo3	Teórico
Grupo3	Práctico
Grupo4	Teórico
Grupo4	Práctico

Comunicación y Dinámica de Grupo

Grupo	Modalidad
Grupo1	Teórico
Grupo1	Práctico
Grupo2	Teórico
Grupo2	Práctico
Grupo3	Teórico

Registro de Docente Por Clase

☐ Por N° de Ciclo ☒ Por Asignatura

Facultad : Ingeniería de Sistemas e Informática

Dir. de Escuela : Ingeniería de Sistemas

Plan de Estudios : Plan 2009

Asignatura : Matemática Básica I

Tipo de Dictado : Teórico

N° de Docente Por Clase : 0

Guardar

Figura 5.6 Mantener Dictado de Clase [Fuente: Elaboración Propia].

5.2.3.1.3 CU- CFH-003: Configurar Periodos de Dictado por Ciclo

Este caso de uso permitirá al Director de Escuela poder actualizar los periodos en que se preferiría que se asignen los dictados de clases según su ciclo como se muestra en la Figura 5.7.

Periodos de Dictado de Asignaturas por Ciclos

Ciclo	Lun-Vie	Sábado
Ciclo I:	Lun-Vie: 8:00 a.m. - 2:00 p.m.	Sábado: 8:00 a.m. - 2:00 p.m.
Ciclo II:	Lun-Vie: 8:00 a.m. - 2:00 p.m.	Sábado: 8:00 a.m. - 2:00 p.m.
Ciclo III:	Lun-Vie: 8:00 a.m. - 3:00 p.m.	Sábado: 8:00 a.m. - 2:00 p.m.
Ciclo IV:	Lun-Vie: 8:00 a.m. - 3:00 p.m.	Sábado: 8:00 a.m. - 2:00 p.m.
Ciclo V:	Lun-Vie: 10:00 a.m. - 7:00 p.m.	Sábado: 8:00 a.m. - 5:00 p.m.
Ciclo VI:	Lun-Vie: 10:00 a.m. - 9:00 p.m.	Sábado: 8:00 a.m. - 5:00 p.m.
Ciclo VII:	Lun-Vie: 2:00 p.m. - 9:00 p.m.	Sábado: 8:00 a.m. - 5:00 p.m.
Ciclo VIII:	Lun-Vie: 2:00 p.m. - 9:00 p.m.	Sábado: 8:00 a.m. - 5:00 p.m.
Ciclo IX:	Lun-Vie: 5:00 p.m. - 9:00 p.m.	Sábado: 8:00 a.m. - 5:00 p.m.
Ciclo X:	Lun-Vie: 5:00 p.m. - 9:00 p.m.	Sábado: 8:00 a.m. - 5:00 p.m.

Guardar

Figura 5.7 Configuración de Periodo de Dictado por Ciclo [Fuente: Elaboración Propia].

5.2.3.1.4 CU- CFH-004: Configurar parámetros del algoritmo

Este caso de uso permitirá al Administrador poder actualizar los parámetros del algoritmo: tamaño de la población, cantidad de iteraciones, la probabilidad de cruzamiento, la probabilidad de mutación, número máximo de horas consecutivas de dictado de clase y número total de horas de clase de asignaturas de un ciclo por día como se muestra en la Figura 5.8.

Configuración de Parámetros

Tamaño de la Población:	<input type="text" value="50"/>
Cantidad de Iteraciones:	<input type="text" value="900"/>
Pc:	<input type="text" value="0.6"/>
Pm:	<input type="text" value="0.3"/>
MaxHorasConsxDia:	<input type="text" value="6"/>
MaxHorasClasexDia:	<input type="text" value="9"/>

Figura 5.8 Configuración de Parámetros del algoritmo [Fuente: Elaboración Propia].

5.2.3.1.5 CU-CFH-005: Gestionar Disponibilidad Horaria

Este caso de uso permitirá al Docente registrar y editar la disponibilidad que podrá dictar para el nuevo ciclo o indicar que su disponibilidad es igual al del ciclo anterior como se muestra en la Figura 5.9.

Disponibilidad Docente

PIEDRA ISUSQUI, JOSE

Periodo	Lunes	Martes	Miércoles	Jueves	Viernes	Sabado
8:00-9:00						
9:00-10:00						
10:00-11:00						
11:00-12:00						
12:00-13:00						
13:00-14:00						
14:00-15:00						
15:00-16:00						
16:00-17:00						
17:00-18:00						
18:00-19:00						
19:00-20:00						
20:00-21:00						
21:00-22:00						

Figura 5.9 Disponibilidad de Docente [Fuente: Elaboración Propia].

5.2.3.1.6 CU-CFH-006: Gestionar Clases de preferencias

Este caso de uso permitirá al Docente registrar y editar las clases que son de su preferencia para que el Jefe de Departamento tenga en consideración al momento de asignarle dictados de clase para un ciclo vigente como se muestra en la Figura 5.10.



Figura 5.10 Clases de preferencia de un Docente [Fuente: Elaboración Propia].

5.2.3.1.7 CU-CFH-007: Asignar Dictado de Clase

Este caso de uso permitirá al Jefe de Departamento asignar los dictados de clases a los docentes como se muestra en la Figura 5.11.

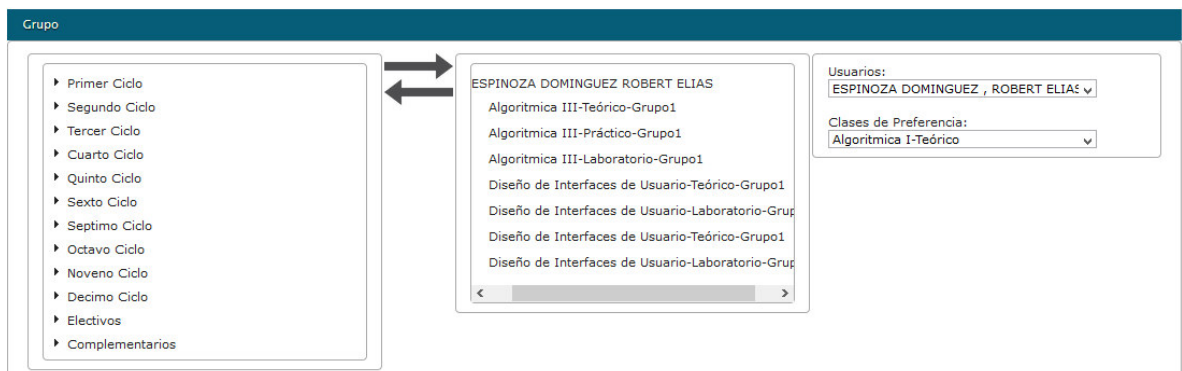


Figura 5.11 Asignar Dictado de Clase [Fuente: Elaboración Propia].

5.2.3.2 Módulo de Seguridad

A través de este subsistema se podrá llevar a cabo el control de acceso a los módulos por los perfiles asociados a los usuarios del sistema, como se muestra en el diagrama casos de usos de la Figura 5.12.

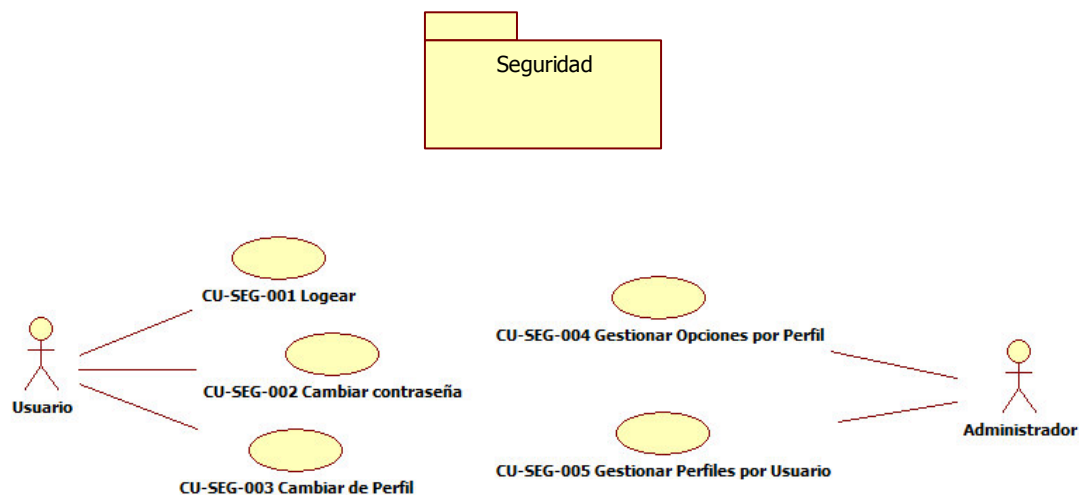


Figura 5.12 Diagrama de Casos de Uso del Módulo Seguridad [Fuente: Elaboración Propia].

El propósito u objetivo de cada caso de uso y la importancia por su impacto en la arquitectura del software se presenta a continuación:

5.2.3.2.1 CU-SEG-001: Loguear Usuario

Este caso de uso es realizado por cualquier usuario que desee ingresar al sistema como se muestra en la Figura 5.13.

Figura 5.13 Loguear Usuario [Fuente: Elaboración Propia].

5.2.3.2.2 CU-SEG-002: Cambiar contraseña

Este caso de uso es realizado por todos los usuarios que desean modificar su contraseña como se muestra en la Figura 5.14.

Cambiar contraseña

Por favor ingrese su nueva contraseña

Nueva Contraseña:

Repita nueva contraseña

Cambiar contraseña

Figura 5.14 Cambiar contraseña [Fuente: Elaboración Propia].

5.2.3.2.3 CU-SEG-003: Cambiar de Perfil

Este caso de uso es realizado por todos los usuarios que desean cambiar de perfil en caso tengan más de 1 perfil asociado como se muestra en la Figura 5.15.

Cambio de Perfil

Perfil x Defecto ☐ Seleccione Perfil:

Actualizar

Figura 5.15 Cambiar de Perfil [Fuente: Elaboración Propia].

5.2.3.2.4 CU-SEG-004: Gestionar Opciones por Perfil

Este caso de uso es realizado por el Administrador quien asignará las opciones del menú a los perfiles como se muestra en la Figura 5.16.

Permisos del usuario

Regresar Registrar

Perfiles: **DOCENTE**

- ☐ Mantenedores
- ☒ Configuración de Horarios
 - ☐ Ciclo
 - ☐ Dictados de Clases
 - ☐ Asignar Clases
 - ☒ Disponibilidad Docente
 - ☐ Conf. Periodo de Dictado
 - ☐ Configurar Parametros
 - ☒ Asignar Carga Academica
- ☒ Elaboración de Horarios
 - ☐ Gestion de Horarios
 - ☒ Visualizar Horarios
- ☒ Seguridad
 - ☐ Perfiles
 - ☐ Usuarios
 - ☒ Cambiar Contraseña
 - ☒ Cambiar Perfil

Figura 5.16 Gestión de Opciones por Perfil [Fuente: Elaboración Propia].

5.2.3.2.5 CU-SEG-005: Gestionar Perfiles por Usuario

Este caso de uso es realizado por el Administrador quien asignará los perfiles que tendrán asociado los usuarios del sistema como se muestra en la Figura 5.17.

Permisos del usuario

Regresar Registrar

Usuarios: **PARIONA QUISPE, JAIME**

- ☒ Perfiles
 - ☐ ADMINISTRADOR
 - ☒ DOCENTE
 - ☐ DIRECTOR DE ESCUELA
 - ☐ JEFE DE DEPARTAMENTO
 - ☐ ASISTENTE DE LOGISTICA
 - ☐ ASISTENTE DE LABORATORIO

Figura 5.17 Gestionar Perfiles por Usuario [Fuente: Elaboración Propia].

5.2.3.3 Módulo de Mantenimiento

A través de este subsistema se podrá llevar a cabo el mantenimiento de las entidades del sistema, según los perfiles de los usuarios podrán registrar, modificar, eliminar y/o consultar como se muestra en el diagrama casos de usos de la Figura 5.18.

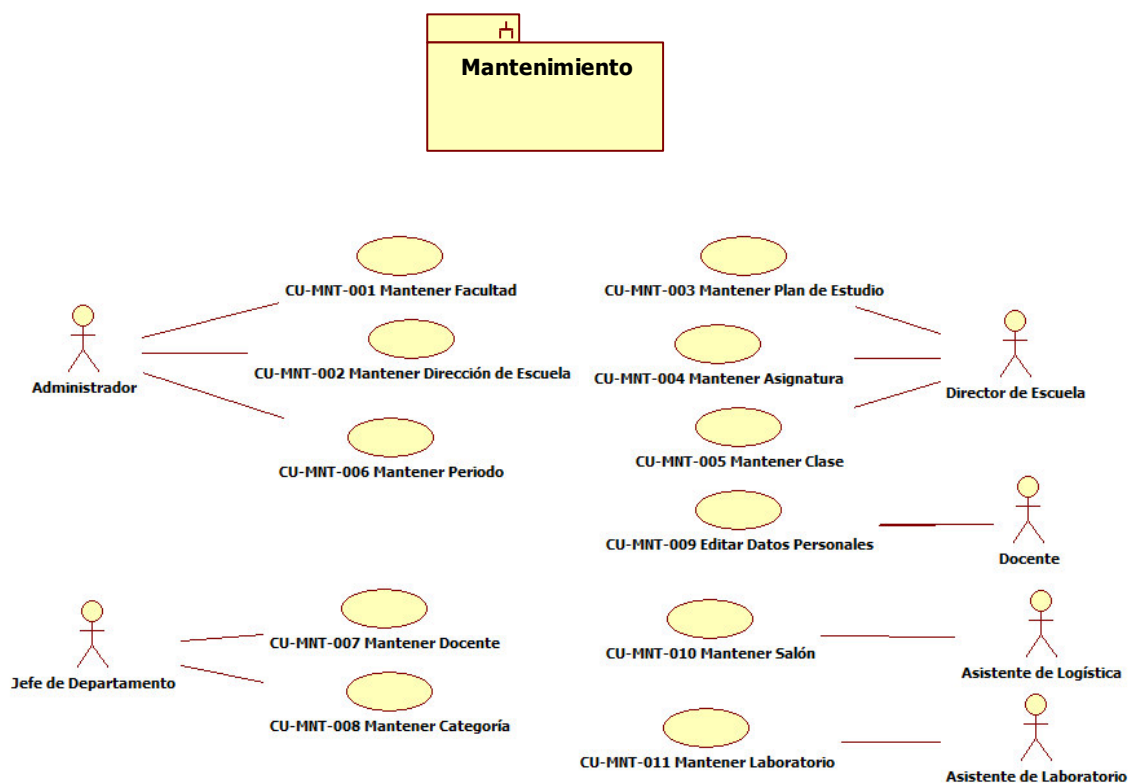


Figura 5.18 Diagrama de Casos de Uso del Módulo Mantenimiento [Fuente: Elaboración Propia].

El propósito u objetivo de cada caso de uso y la importancia por su impacto en la arquitectura del software se presenta a continuación:

5.2.3.3.5 CU-MNT-001: Mantener Facultad

Este caso de uso es realizado por el Administrador para consultar, registrar, actualizar y eliminar una Facultad.

5.2.3.3.2 CU-MNT-002: Mantener Dirección de Escuela

Este caso de uso es realizado por el Administrador para consultar, registrar, actualizar y eliminar una Dirección de Escuela como se muestra en la Figura 5.19.

Dir. de Escuela				
Universidad	Facultad	Dir. de Escuela	Fecha de Creación	
	Ingeniería de Sistemas e Informática	Ingeniería de Sistemas	2002-01-01	
	Ingeniería de Sistemas e Informática	Ingeniería de Software	2009-02-01	

Figura 5.19 Mantenimiento de Dirección de Escuela [Fuente: Elaboración Propia].

5.2.3.3.3 CU-MNT-003: Mantener Plan de Estudios

Este caso de uso es realizado por el Director de Escuela para consultar, registrar, actualizar y eliminar un Plan de Estudios como se muestra en la Figura 5.20.

Planes de Estudio					
Universidad	Facultad	Dir. Escuela	Plan Estudios	Fecha Creación	
	Ingeniería de Sistemas e Informática	Ingeniería de Sistemas	Plan 2002	01-01-2002	
	Ingeniería de Sistemas e Informática	Ingeniería de Sistemas	Plan 2009	01-01-2009	
	Ingeniería de Sistemas e Informática	Ingeniería de Sistemas	Plan 2014	01-01-2014	

Registrar

Figura 5.20 Mantenimiento de Plan de Estudios [Fuente: Elaboración Propia].

5.2.3.3.4 CU-MNT-004: Mantener Asignatura

Este caso de uso es realizado por el Director de Escuela para consultar, registrar, actualizar y eliminar una Asignatura como se muestra en la Figura 5.21.

Lista de Asignaturas - Plan de estudios : Plan 2009						
Seleccione Plan de Estudios: Plan 2009						
Id	Codigo	Ciclo	Nombre	Cred.		
1	201001	CICLO I	Algoritmica I	4		
2	201008	CICLO I	Computacion e Informatica	4		
3	201003	CICLO I	Calculo I	4		
4	201004	CICLO I	Matematica Basica I	4		
5	201007	CICLO I	Teoria de Sistemas	3		
6	201009	CICLO I	Taller de Tecnicas de Estudio	3		
7	202001	CICLO II	Algoritmica II	4		
8	202008	CICLO II	Estructura de Datos	4		
9	202003	CICLO II	Calculo II	4		
10	202007	CICLO II	Fisica General	4		
11	202004	CICLO II	Matematica Basica II	4		
12	202009	CICLO II	Economia	2		
13	204001	CICLO III	Algoritmica III	4		
14	204006	CICLO III	Diseño Grafico	3		
15	204005	CICLO III	Estadística I	4		

Registrar

Figura 5.21 Mantenimiento de Asignaturas [Fuente: Elaboración Propia].

5.2.3.3.5 CU-MNT-005: Mantener Clase

Este caso de uso es realizado por el Director de Escuela para consultar, registrar, actualizar y eliminar una Clase como se muestra en la Figura 5.22.

Lista de Clases - Plan de estudios : Plan 2009

Seleccione Plan de Estudios: Plan 2009

Ciclo	Asignatura	Tipo de Dictado	Nº de horas	Capacidad	Estado		
CICLO I	Algoritmica I	Teórico	2.00 h.	40	1		
CICLO I	Algoritmica I	Práctico	2.00 h.	40	1		
CICLO I	Algoritmica I	Laboratorio	2.00 h.	20	1		
CICLO I	Computacion e Informatica	Teórico	3.00 h.	40	1		
CICLO I	Computacion e Informatica	Laboratorio	2.00 h.	20	1		
CICLO I	Calculo I	Teórico	3.00 h.	40	1		
CICLO I	Calculo I	Práctico	2.00 h.	40	1		
CICLO I	Matematica Basica I	Teórico	3.00 h.	40	1		
CICLO I	Matematica Basica I	Práctico	2.00 h.	40	1		
CICLO I	Teoria de Sistemas	Teórico	3.00 h.	30	1		
CICLO I	Taller de Tecnicas de Estudio	Teórico	3.00 h.	20	1		
CICLO II	Algoritmica II	Teórico	2.00 h.	40	1		
CICLO II	Algoritmica II	Práctico	2.00 h.	40	1		
CICLO II	Algoritmica II	Laboratorio	2.00 h.	20	1		
CICLO II	Estructura de Datos	Teórico	3.00 h.	30	1		
CICLO II	Estructura de Datos	Práctico	2.00 h.	30	1		
CICLO II	Calculo II	Teórico	3.00 h.	40	1		
CICLO II	Calculo II	Práctico	2.00 h.	40	1		
CICLO II	Fisica General	Teórico	2.00 h.	40	1		
CICLO II	Fisica General	Práctico	2.00 h.	40	1		

1 2 3 4 5 6 7

Registrar

Figura 5.22 Mantenimiento de Clases [Fuente: Elaboración Propia].

5.2.3.3.6 CU-MNT-006: Mantener Periodo

Este caso de uso es realizado por el Administrador para consultar, registrar, actualizar y eliminar Periodos.

5.2.3.3.7 CU-MNT-007: Mantener Docente

Este caso de uso es realizado por el Jefe de Departamento para consultar, registrar, actualizar y cesar un Docente como se muestra en la Figura 5.23.

Lista de Docentes					
Codigo	Nombre	Apellidos	Email		
91081	MARIA ROSA	DAMASO RIOS			
09831E	ROBERT ELIAS	ESPINOZA DOMINGUEZ			
2313	FLORIAN LAZARO	MOTA ALVA			
	JAIME	PARIONA QUISPE			
95699	PABLO	ROMERO NAUPARI			
97543	NORBERTO	OSORIO BELTRAN			
95702	JAVIER ARTURO	GAMBOA CRUZADO			
10162	LUIS	GUERRA GRADOS			
10936	LUZMILA	PRO CONCEPCION			
11797	GILBERTO	SALINAS AZAÑA			
0A1234	MELVA	VALVERDE AYALA			
09434A	CARLOS	CANEPA PEREZ			
95729	JUAN CARLOS	GONZALES SUAREZ			
17779	DAVID	MAURICIO SANCHEZ			
0A1232	NEHIL	MUÑOZ CASILDO			

Figura 5.23 Mantenimiento de Docentes [Fuente: Elaboración Propia].

5.2.3.3.8 CU-MNT-008: Mantener Categoría

Este caso de uso es realizado por el Jefe de Departamento para consultar, registrar, actualizar y eliminar una Categoría.

5.2.3.3.9 CU-MNT-009: Mantener Salón

Este caso de uso es realizado por el Asistente de Logística para consultar, registrar, actualizar y eliminar un Salón como se muestra en la Figura 5.24.

Lista de Salones					
Nombre	Nº de Piso	Capacidad	Estado		
101	1	42	Disponible		
103	1	30	Disponible		
103A	1		Disponible		
104	1		Disponible		
105A	1		Disponible		
106	1	36	Disponible		
108	1		Disponible		
200-B	1		Disponible		
201	1		Disponible		
202	1		Disponible		

Figura 5.24 Mantenimiento de Salones [Fuente: Elaboración Propia].

5.2.3.3.10 CU-MNT-010: Mantener Laboratorio

Este caso de uso es realizado por el Asistente de Logística para consultar, registrar, actualizar y eliminar un Laboratorio como se muestra en la Figura 5.25.

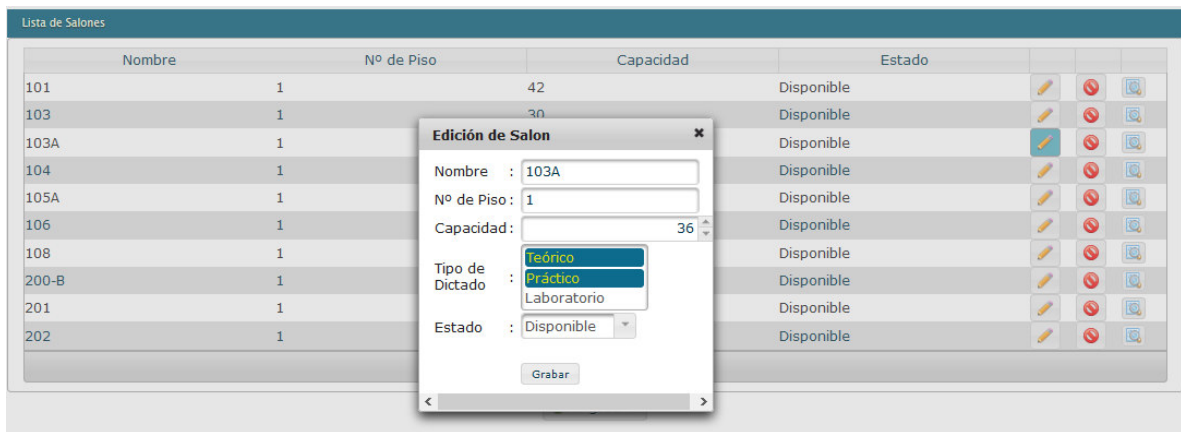


Figura 5.25 Mantenimiento de Laboratorios [Fuente: Elaboración Propia].

5.2.3.4 Módulo de Elaboración de Horarios

A través de este subsistema se podrá consultar y generar horarios para un ciclo académico, además podrá obtener diferentes tipos de reportes: horarios por salón, horario por ciclo y uno más detallado las clases por salón y por periodo como se muestra en el diagrama casos de usos de la Figura 5.26.

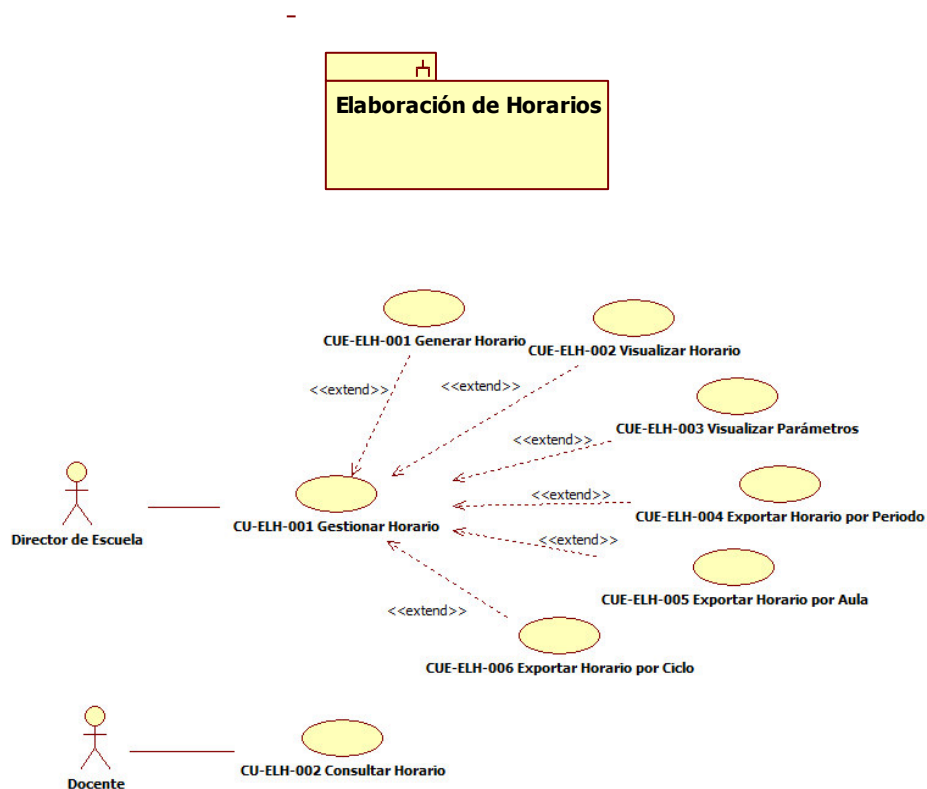


Figura 5.26 Diagrama de Casos de Uso del Módulo Elaboración de Horarios [Fuente: Elaboración Propia].

El propósito u objetivo de cada caso de uso y la importancia por su impacto en la arquitectura del software se presenta a continuación:

5.2.3.4.10 CU-ELH-001: Gestión de Horarios

Este caso de uso es realizado por el Director de Escuela quien puede iniciar la generación de un horario con los datos previos de la configuración del horario, podrá consultar los horarios que se hayan generado y podrá generar los reportes: Salón por periodo, Horarios por salón y Horario por ciclo como se muestra en la Figura 5.27.



Código	Función de Aptitud
1	2043
19	261
20	227
21	641
22	177
23	218
24	285
25	755
26	530
27	162

Figura 5.27 Gestión de Horarios [Fuente: Elaboración Propia].

5.2.3.4.11 CU- ELH-002: Consultar Horario

Este caso de uso es realizado por el Docente quien puede consultar el horario asignado para el ciclo vigente como se muestra en la Figura 5.28.

Horario Docente						
PIEDRA ISUSQUI, JOSE						
Periodo	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
8:00-9:00						Gerencia Informatica - T - 105A
9:00-10:00						Gerencia Informatica - T - 105A
10:00-11:00						Gerencia Informatica - T - 105A
11:00-12:00						Gerencia Informatica - T - 105A
12:00-13:00						
13:00-14:00						
14:00-15:00		Ingenieria de Informacion - T - 105A				
15:00-16:00		Ingenieria de Informacion - T - 105A				
16:00-17:00		Ingenieria de Informacion - T - 105A				
17:00-18:00				Ingenieria de Negocios - T - 202		
18:00-19:00			Ingenieria de Informacion - P - 202	Ingenieria de Negocios - T - 202		
19:00-20:00			Ingenieria de Informacion - P - 202			
20:00-21:00						
21:00-22:00						

Figura 5.28 Consultar Horario [Fuente: Elaboración Propia].

5.3 Análisis y Diseño

En esta disciplina se transforman los requerimientos en un diseño de sistema, se desarrolla una arquitectura robusta y se adapta el diseño para concordar con el ambiente de implementación.

Los principales artefactos que se relacionan con esta disciplina se encuentran especificados en el Anexo 2.

5.4 Arquitectura

La arquitectura es lo que especifica el arquitecto en la descripción de la arquitectura. La descripción de la arquitectura permite controlar el desarrollo el sistema desde la perspectiva técnica. La descripción de la arquitectura es una vista de los modelos del sistema, de los modelos de casos de uso, análisis, diseño, implementación y despliegue. La descripción de la arquitectura describe las partes del sistema que es importante que comprendan todos los desarrolladores y otros interesados. En el anexo 3 se describe el documento de Arquitectura.

Capítulo 6: Estudio Experimental

Para realizar el estudio experimental de la solución del problema de programación de horarios mediante el algoritmo genético, se explicará cómo se organizan las Facultades, se realizará la prueba de análisis de sensibilidad de los parámetros y la comparación con un caso de estudio.

6.1 Organización de la Facultad

La actividad académica en las Facultades se organiza en carreras, años, ciclos, asignaturas, grupos y clases.

6.1.1 Carreras, Años y Ciclos

Cada carrera, por ejemplo: Ingeniería Civil, Ingeniería Mecánica, Ingeniería en Computación, etc., tiene una duración entre 4 a 6 años. A su vez cada año se divide en uno, dos o tres ciclos; esto dependerá específicamente de como se ha elaborado la malla curricular en la Universidad.

6.1.2 Asignaturas

Una asignatura o curso puede pertenecer a más de una carrera y no necesariamente al mismo año dentro de ellas, pudiendo además tener una identificación diferente dependiendo de la carrera. Esas asignaturas comunes a las carreras son agrupadas en conjuntos de asignaturas equivalentes y se identifica una asignatura primaria, que es la única que se considera en representación de las otras para la asignación.

Las asignaturas tienen diferentes niveles de importancia, los cuales determinan las prioridades a la hora de establecer la asignación. Por ejemplo, las asignaturas del ciclo básico tienen mayor prioridad que las de las otras carreras y dentro de una misma carrera son más importantes las asignaturas obligatorias que las electivas. Por lo tanto se definen los siguientes niveles en orden de importancia:

- **Básicas:** Asignaturas pertenecientes al ciclo básico de la Facultad de Ingeniería, deben tener precedencia al momento de asignarlas.

- **Obligatorias:** Asignaturas que son obligatorias para las carreras a las cuales pertenece.
- **Electivas:** Asignaturas electivas para alguna o todas las carreras a las cuales pertenece, su importancia dentro de la asignación es menor a la de las obligatorias.

Debido a que las asignaturas pueden pertenecer a más de una carrera, el criterio de importancia es una característica de la relación entre una asignatura y una carrera, ya que una misma asignatura puede ser obligatoria para una carrera y electiva para otra. Además, se debe de considerar un límite de alumnos para algunos cursos de una malla curricular, con ello restringimos la capacidad del aula.

Generalmente, se debe tener estimados para cada asignatura los porcentajes de abandono y recursantes dentro de los alumnos inscritos. Es importante tener en cuenta esos porcentajes al realizar la asignación para determinar si la capacidad de los salones es suficiente para la cantidad de alumnos que efectivamente concurrirán a esa asignatura.

6.1.3 Grupos

Las asignaturas se organizan en uno o más grupos, entre los cuales se distribuyen los alumnos inscritos a las mismas. Los grupos pueden ser de teoría, práctica o de laboratorio y cada asignatura tiene al menos un grupo.

6.1.4 Clases

Cada grupo de cada asignatura es dividido en clases. Una clase es la unidad básica a asignar a los salones y horarios. Cada clase tiene una duración, días y horarios de preferencia. La cantidad de alumnos que asisten a una clase queda determinada por la cantidad de alumnos asignados al grupo al cual pertenece.

Existen asignaciones de clase que deben ser fijas, es decir que el proceso de búsqueda de una asignación no puede modificar. Una clase puede necesitar que el salón en que se dicta disponga de cierta infraestructura con diferentes recursos, como por ejemplo enchufes, proyector, CPU, pizarra, etc.

6.1.5 Turnos

Debido a la gran cantidad de estudiantes que ingresan en primer año se dividen el dictado de las clases de los grupos en turnos diferentes y a criterio del elaborador de la programación horaria. Para cada clase se determina cual es el turno que le corresponde. Se definen los siguientes turnos:

- ✓ Mañana
- ✓ Tarde
- ✓ Noche
- ✓ Todo el día, para las asignaturas que no tengan preferencias horarias.
- ✓ Un horario que incluya periodos específicos.

6.1.6 Salones

La principal característica que se tiene en cuenta de los salones es su capacidad. También se tiene en cuenta las características de infraestructura (recursos) y datos informativos como el nombre y la ubicación.

Existen asignaciones de clases a horarios que son fijas, por lo cual no deben modificarse en el proceso de asignación. Del mismo modo, existen salones que no se pueden utilizar libremente, y no se pueden tomar en cuenta en la asignación automática.

Se definen tres estados posibles para los salones:

- **Asignación Libre:** Salón en uso, disponible para el proceso automático de asignación.
- **Asignación Limitada:** Salón en uso, solo utilizable en asignaciones manuales, por ejemplo los salones pertenecientes a las distintas facultades, los cuales son asignados con un permiso.
- **No Disponible:** Salón en desuso, no disponible para el proceso automático de asignación, ni para asignaciones manuales.

6.1.7 Características

Existen clases que requieren de cierta infraestructura para que su dictado sea posible para ello se debe asignar un salón que tenga las características necesarias. Por ejemplo la capacidad y si el salón es adecuado para un tipo de dictado.

6.1.8 Horarios de dictado de Clases

Las clases se dictan desde las 8:00 a las 22:00 horas de lunes a sábado y la unidad mínima de dictado es 1 hora. Por lo tanto la duración de cada clase se expresa en múltiplos de la misma.

En caso particular, para la Facultad de Ingeniería de Sistemas de la Universidad Nacional Mayor de San Marcos, el diagrama de procesos que actualmente se realiza manualmente como se muestra en la Figura 6.1:

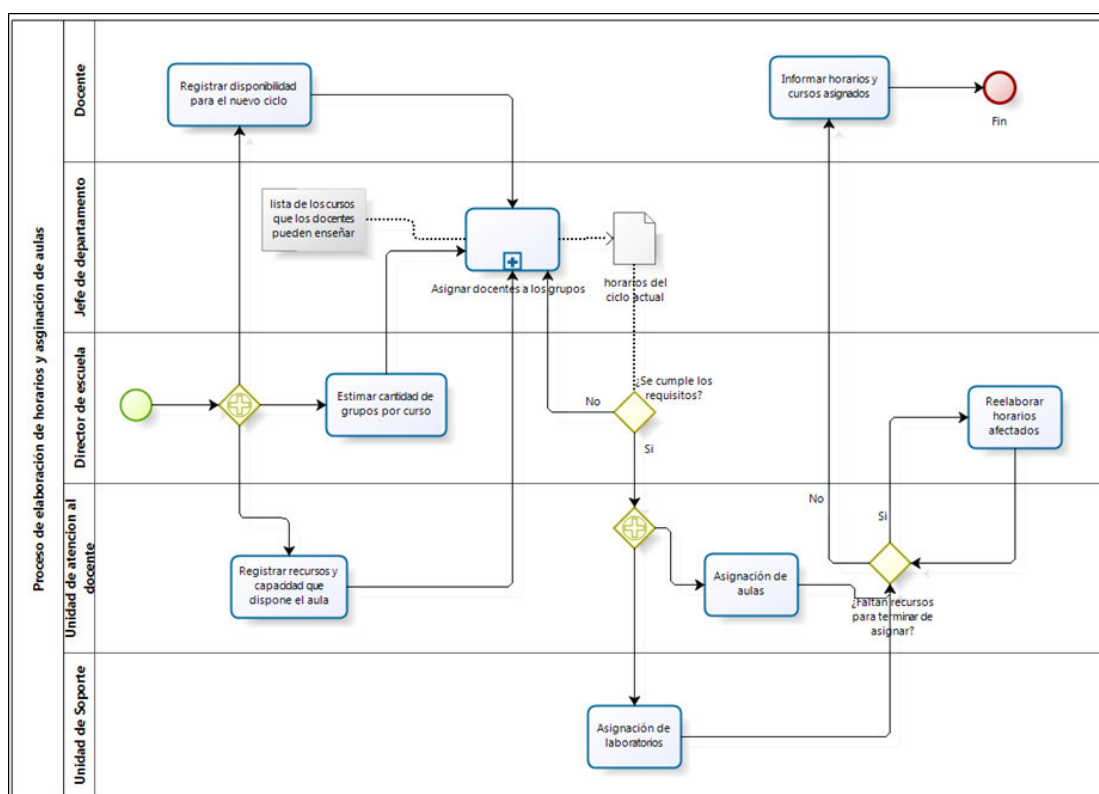


Figura 6.1 Diagrama de procesos de la FISI [Fuente: Elaboración Propia].

En el caso de la FISI los inconvenientes que se presentan en la elaboración de horarios son los siguientes:

La asignación de docentes a los grupos demora al director de escuela y al jefe de departamento académico más de 3 semanas.

Hay docentes que cambian su disponibilidad fuera de tiempo y hay que evaluar nuevamente el horario de los grupos asignados con la disponibilidad modificada.

No se considera los recursos (aulas y laboratorios) con los que cuenta la FISI al momento de cuadrar horarios de docentes con cantidad de grupos, lo que ocasiona que no hay recursos libres para asignar el curso en ese horario y que conduce a reprogramar ese horario.

La publicación de los horarios se hace pocos días antes de la matrícula y no puede haber feedback, por lo cual los alumnos encuentran cruces de horarios por grupo además de que se sobrecarga en horas por día lo cual perjudica al alumno teniendo cruces de 1 o más horas durante todo el ciclo.

6.2 Análisis de Sensibilidad de los Parámetros

Con el fin de descubrir los valores de los parámetros que tienen gran efecto sobre el rendimiento y de los resultados del algoritmo se aplicarán dos tipos de pruebas:

6.2.1 Prueba de Número de Generaciones

Existen investigaciones en las cuales se han modificado la variable número de generaciones, con el objetivo de conocer el comportamiento de las soluciones a través de un número de generaciones dado.

6.2.2 Prueba de Porcentajes

El objetivo es conocer el efecto que tienen los operadores genéticos en la función de adaptación. Consiste en cambiar el porcentaje de ocurrencia de los operadores genéticos.

Estas 2 pruebas se realizarán en una misma corrida. En la Tabla 6.1 los diferentes valores establecidos de los parámetros para el experimento.

Pm	Pc	Tamaño de la Población	Cantidad de Iteraciones
0.1	0.6	50	300
0.2	0.7	70	500
0.3	0.8	90	700
0.4	0.9	120	900

Tabla 6.1 Valores de los parámetros del algoritmo [Fuente: Elaboración Propia].

El total de combinaciones de todos los valores de los 4 parámetros es de 256, cuya ejecución en el algoritmo nos brindará para cada combinación la función de aptitud y el tiempo requerido para generar el horario. Después de analizar los valores obtenidos del total de las combinaciones del Anexo I, se destacan las 12 mejores combinaciones que han tenido la menor función de aptitud y en un tiempo menor, como se muestra la tabla 6.2.

Nº	Pm	Pc	Tamaño de la población	Cantidad de Iteraciones	Función de Aptitud	Tiempo (min)
1	0.3	0.7	50	900	197	9.80
2	0.1	0.8	50	900	198	9.85
3	0.3	0.6	70	900	204	9.73
4	0.4	0.6	110	700	207	8.33
5	0.4	0.6	50	700	210	7.75
6	0.1	0.6	50	700	211	7.88
7	0.4	0.7	90	900	211	11.27
8	0.3	0.7	110	700	212	8.08
9	0.4	0.7	50	900	212	9.78
10	0.1	0.6	110	900	212	11.28
11	0.2	0.7	50	900	213	9.65
12	0.4	0.6	50	900	214	9.82

Tabla 6.2 Las 12 mejores combinaciones de parámetros [Fuente: Elaboración Propia].

6.3 Caso de Pruebas

En esta sección se realiza una comparación mediante indicadores de los resultados obtenidos entre el sistema propuesto y la elaboración manual de horarios.

6.3.1 Descripción de la Instancia

La información empleada en las pruebas, será obtenida del horario de la Escuela de Ingeniería de Sistemas de la Facultad de Ingeniería Sistemas e Informática de la Universidad Nacional Mayor de San Marcos para el semestre académico 2015-I. Los datos de prueba son los siguientes:

1. Se cuenta con un total de 55 asignaturas del plan de estudios 2009 y 18 asignaturas del plan de estudios 2014.
2. Se cuenta con un total de 125 docentes de las diferentes categorías.
3. Se cuenta con un total de 27 salones, de las cuales 7 son laboratorios y 20 son de clases teóricas y prácticas. Adicionalmente hay 2 laboratorios no disponibles debido a que se encuentran reservados para otros eventos y para la generación del horario se considera la disponibilidad de los salones durante la semana.
4. Cada asignatura tiene una cantidad de clases las cuales pueden ser de teoría, de práctica o de laboratorio, para las 73 asignaturas hay en total 134 clases.
5. El Director de Escuela crea una cierta cantidad de secciones o grupos por cada clase y para el semestre académico 2015-I se ha creado 380 secciones.
6. La cantidad de periodos a considerar es de 84 periodos de tiempo posibles que van desde el lunes a sábado desde las 8 am hasta las 10:00 pm y la duración de cada periodo es de 60 minutos como se muestra en la Figura 6.2:

Periodo 1	Periodo 2		Periodo 84
Lunes 8am-9am	Lunes 9am-10am	...	Sábado 9pm-10pm

Figura 6.2 Periodos de Clase [Fuente: Elaboración Propia].

6.3.2 Criterios de Comparación

El objetivo central es mostrar las ventajas de la utilización del sistema propuesto, para esto se comparan distintos indicadores de desempeño. Estos indicadores fueron obtenidos después de evaluar cada horario en la función de aptitud. Los indicadores utilizados fueron los siguientes:

- I1: Número de docentes que tienen asignado más de un dictado de clase en el mismo periodo.
- I2: Número de asignaciones de dictados de clases en periodos no disponibles para un docente.
- I3: Número de dictados de clases que solapan con dictados de clases que no pueden ser asignadas en el mismo periodo.
- I4: Número de dictados de clases con horario fuera del rango de periodos establecidos por ciclo.
- I5: Número de horas excedentes a 6 periodos consecutivos de dictados de clases asignadas en un mismo día perteneciente a un ciclo.
- I6: Número de horas excedentes a 9 periodos de dictados de clases asignadas en un mismo día perteneciente a un ciclo.
- I7: Número de horas de separación entre dictados de clases de un mismo grupo en un mismo día perteneciente a un mismo ciclo por día.
- I8: Tiempo de generación de la programación.

6.4 Análisis de Resultados

La solución entregada por el sistema actual no cumple con las condiciones deseables para la operatividad normal de la institución. La Tabla 6.3 muestra los resultados obtenidos para los horarios obtenidos por el sistema manual y el sistema propuesto.

Indicador	Horario Manual	Horario del Sistema
I1	0	0
I2	68	0
I3	14	0
I4	138	79
I5	254	77
I6	26	7
I7	82	34
I8	1 mes	9.80 min

Tabla 6.3 Medición de indicadores para los horarios 2015-I [Fuente: Elaboración Propia].

En el indicador I1 se basa en que un docente no puede dictar más de un dictado de clase en el mismo periodo, como se aprecia el horario propuesto, no incumple la restricción.

En el indicador I2, se puede observar que la disponibilidad de docente se respeta, este indicador es muy importante para que los docentes no tengan conflictos por tener asignado dictados de clases en periodos en el que no pueda asistir.

En el indicador I3, el sistema reduce la asignación de clases con otras clases que tengan conflicto debido a que es importante evitar dichos cruzamientos.

En el indicador I4, el sistema permitirá una mayor dispersión de las clases para que los estudiantes tengan más posibilidades de escoger cualquier sección de clase sin que se cruce con otra clase del mismo ciclo.

Los indicadores I5 y I6 se reducen a un cierto valor mínimo, este factor nos indica que los alumnos no tendrán posibilidad de sobrecarga de clases con más de 6 horas consecutivas o más de 9 horas de clases por día.

En el indicador I7, el sistema reduce las horas de separación entre clases de un mismo grupo perteneciente a un mismo ciclo, ya que se desea que el dictado de dichas clases sean continuas y se evite el tiempo de espera por parte de los alumnos.

Respecto a las condiciones básicas del sistema, el algoritmo propuesto asigna todas las clases a un periodo y salón, no permite que una clase sea asignada a aulas no factibles, no permite la asignación de dos o más cursos en un mismo salón y genera siempre un horario con todas las clases que se desea programar.

El sistema actual viola estas condiciones, ya que después de publicado el horario se crean más clases debido a que antes no se podía agregar más clases porque no cumplían con las restricciones establecidas.

Finalmente un indicador muy importante a la hora de definir la programación de clases y la asignación de salones es el tiempo que demora dicho proceso. Las mejoras en este sentido son sustanciales, pasando desde un mes para el procedimiento manual a 9.37 minutos con el sistema propuesto.

Esta mejora permite no solo obtener una solución óptima en un corto tiempo, sino que es posible explorar múltiples escenarios, obtener soluciones alternativas y poder reaccionar ante eventos inesperados, como lo son la incorporación de nuevas clases

sobre la marcha.

En resumen, los beneficios reportados por el sistema propuesto son evidentes.

La utilización de este sistema permitirá a la Facultad de Ingeniería de Sistemas e Informática eliminar todos los conflictos y satisfacer todas las condiciones de operación, y además, ser eficiente en la asignación de un recurso escaso como son los salones.

Adicionalmente se muestra en la Figura 6.3 la comparación de la Función de Aptitud del horario manual con el horario obtenido del sistema, se puede observar la gran diferencia que existe con la violación de restricciones de ambos horarios.

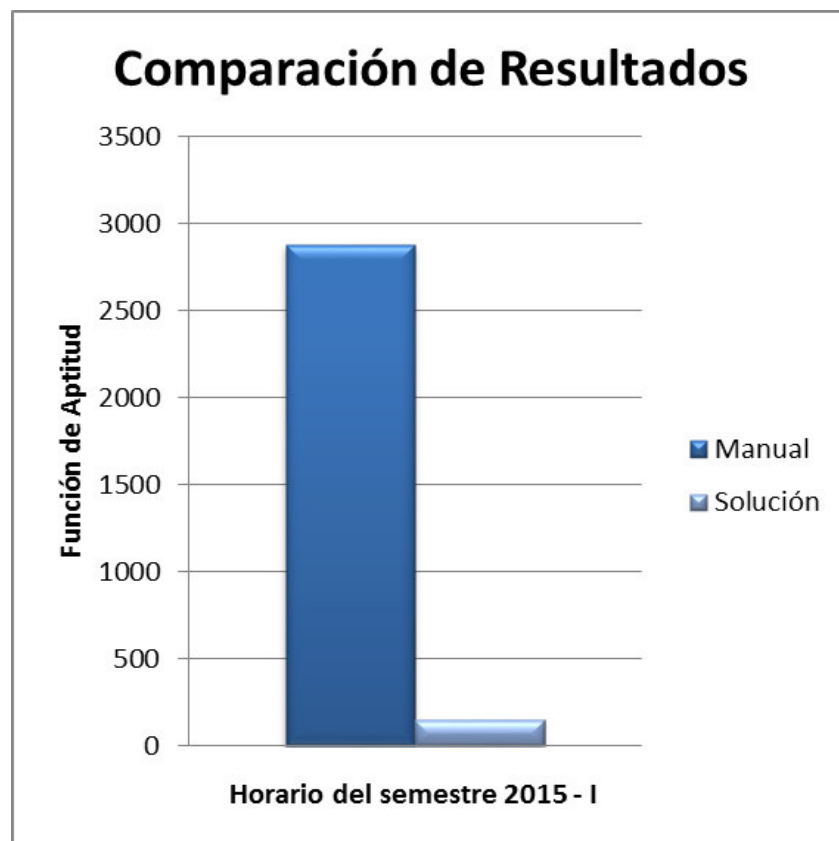


Figura 6.3 Comparación del número de violaciones duras y blandas entre el horario manual y el algoritmo propuesto [Fuente: Elaboración Propia].

Capítulo 7: Conclusiones y Trabajos Futuros

7.1 Conclusiones

En este trabajo se presentó un sistema de generación de horarios para resolver el problema de programación de horarios tomando como caso de estudio la Facultad de Ingeniería de Sistemas e Informática en la Universidad Nacional Mayor de San Marcos en donde llegamos a las siguientes conclusiones:

C1: Se realizó una revisión de la literatura de los algoritmos para el problema de generación de horarios universitarios, observándose que la técnica basada en algoritmos genéticos presenta mejores resultados que otros algoritmos de la literatura como se vio en la Tabla 2.6.

C2: Se establecieron los mejores valores para los parámetros del algoritmo como se detalló en la Tabla 6.2, después de haber realizado pruebas al algoritmo con una combinación de valores como se mostró en la Tabla 6.1.

C3: Se documentaron los resultados obtenidos con el algoritmo propuesto y se realizó la comparación con el horario generado de forma manual para el Ciclo 2015-I, el cual nos arrojó un horario factible, es decir, no hubo violaciones de restricciones obligatorias y se mejoró en forma significativa el cumplimiento de las condiciones deseables a diferencia del horario manual que no fue un horario factible, como se mostró en la Figura 6.3, lo cual genera que se tengan que hacer cambios en los horarios para obtener un horario factible.

C4: Se determinó que el tiempo promedio para la generación de horarios después de cargada la información necesaria en el sistema es de 10 minutos (según la cantidad de dictados de clase y recursos a asignar puede aumentar), comparado con el horario manual el cual tiene un promedio de 1 mes para su elaboración, como se mostró en la Tabla 6.3, y está sujeto a errores humanos, por lo cual resulta una alternativa muy conveniente para su implementación.

7.2 Trabajos Futuros

Dentro de los trabajos futuros se está considerando:

- T 1. Buscar nuevos operadores genéticos avanzados y nuevas heurísticas de los cuales se puedan aprovechar sus beneficios y ayuden a mejorar el algoritmo propuesto, en reducir el número de violaciones de restricciones blandas.
- T 2. Implementar un operador de asignación de salones a dictados de clase que considere sus capacidades de manera óptima y así poder tener más disponibilidad de salones con mayor capacidad.
- T 3. Implementar nuevas características para que el sistema sea más configurable y extensible. Extensible en cuanto a su uso para otras universidades y a si no solamente para las Facultades que tengan restricciones iguales a la Facultad de Sistemas si no para otras Universidades. Con la finalidad de hacer el software más configurable, se recomienda permitir a los futuros usuarios poder escoger entre todas las restricciones que maneja el sistema y decidir cuáles se acoplan a su realidad o problema específico.

Referencias Bibliográfica

- [1]. Shengxiang Yang and Sadaf Naseem Jat, Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 41, NO. 1 (2011) 93-106.
- [2]. Mohammed Azmi Al-Betar , Ahamad Tajudin Khader, Osama Muslih, Al-zaytoonah University of Jordan, Amman, Jordan, A MultiSwap Algorithm for the University Course Timetabling Problem, IEEE-International Conference on Computer & Information Science (2012) 301-306.
- [3]. Salwani Abdullah , Hamza Turabieh, On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems, Elsevier, 191 (2012) 146–168.
- [4]. Mohammed Azmi Al-Betar , Ahamad Tajudin Khader A harmony search algorithm for university course timetabling Annals of Operations Research, April 2012, Volume 194, Issue 1, pp 3-31.
- [5]. Abdullah, S., Turabieh, H., McCollum, B., & McMullan, P. (2012). A hybrid metaheuristic approach to the university course timetabling problem. Journal of Heuristics, 18(1), 1–23.
- [6]. K. Socha, J. Knowles, and M. Samples, "A max-min ant system for the university course timetabling problem," in Proceedings of the 3rd International Workshop on Ant Algorithms, ANTS 2002, ser. LNCS, M. D. et al., Ed., vol. 2463. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 1- 13.
- [7]. S. N. Jat and S. Yang, "A guided search genetic algorithm for the university course timetabling problem," in Proc. 4th Multidisciplinary Int. Conf. Scheduling: Theory Appl., 2009, pp. 180–191.
- [8]. Tri A. Budiono, Kok Wai Wong, A Pure Graph Coloring Constructive Heuristic in Timetabling, IEEE-International Conference on Computer & Information Science (2012) 307-312.
- [9]. C. C. Gotlieb, "The construction of class-teacher timetables," in Proc. IFIP Congr., C. M. Popplewell, Ed., 1962, pp. 73–77.
- [10]. M. A. Al-Betar, A. T. Khader, and A. T. Gani, "A harmony search algorithm for university course timetabling," in Proc. 7th Int. Conf. Pract. Theory Automated Timetabling, 2008, pp. 1–12.
- [11]. M. Tuga, R. Berretta, and A. Mendes, "A hybrid simulated annealing with kempe chain neighborhood for the university timetabling problem," in Proc. 6th IEEE/ACIS Int. Conf. Comput. Inf. Sci., 2007, pp. 400–405.
- [12]. S. Abdullah, E. K. Burke, and B. McCollum, "Using a randomized iterative improvement algorithm with composite neighbourhood structures," in Proc. 6th Int. Conf. Metaheuristic, 2007, pp. 153–169.

- [13]. Z. L"u and J. K. Hao, "Adaptive tabu search for course timetabling," *Eur.J. Oper. Res.*, vol. 200, no. 1, pp. 235–244, Jan. 2010.
- [14]. S. F. H. Irene, S. Deris, and S. Z. M. Hashim, "A study on PSO-based university course timetabling problem," in *Proc. Int. Conf. Adv. Comput. Control*, 2009, pp. 648–651.
- [15]. K. Socha, J. Knowles, and M. Samples, "A max-min ant system for the university course timetabling problem," in *Proc. 3rd Int. Workshop Ant Algorithms (Lecture Notes in Computer Science)*, vol. 2463, 2002, pp. 1–13.
- [16]. M. R. Malim, A. T. Khader, and A. Mustafa, "Artificial immune algorithms for university timetabling," in *Proc 6th Int. Conf. Pract. Theory Automated imetabling*, 2006, pp. 234–245.
- [17]. B. McCollum, "University timetabling: Bridging the gap between research and practice," in *Proc. 6th Int. Conf. Pract. Theory Automated Timetabling*, 2006, pp. 15–35.
- [18]. P. Pongcharoen, W. Promtet, P. Yenradee, and C. Hicks, "Schotastic ptimization timetabling tool for university course scheduling," *Int. J. Prod. Econ.*, vol. 112, no. 2, pp. 903–918, Apr. 2008.
- [19]. D. Landa-Silva and J. H. Obit, "Great deluge with non-linear decay rate for solving course timetabling problems," in *Proc. 4th IEEE Int. Conf. Intell. Syst.*, 2008, pp. 811–818.
- [20]. S. Abdullah and H. Turabieh, "Generating university course timetable using genetic algorithm and local search," in *Proc. 3rd Int. Conf. Hybrid Inform. Tech.*, 2008, pp. 254–260.
- [21]. S. Abdullah, E. K. Burke, B. McCollum, and H. Turabieh, "A hybrid evolutionary approach to the university course timetabling problem," in *Proc. 2007 Congr. Evol. Comput.*, 2007, pp. 1764–1768.
- [22]. E. K. Burke, B. MacCloumn, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for educational timetabling problems," *Eur. J. Oper. Res.*, vol. 176, no. 1, pp. 177–192, Jan. 2007.
- [23]. E. K. Burke, G. Kendall, and E. Soubeiga, "A tabu-search hyper-heuristic for timetabling and rostering," *J. Heuristics*, vol. 9, no. 6, pp. 451–470, 2003.
- [24]. O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. St"utzle, "A comparison of the performance of different metaheuristics on the timetabling problem," in *Proc. 4th Int. Conf. Pract. Theory Automated Timetabling (Lecture Notes in Computer Science)*, vol. 2740, 2003, pp. 329–351.
- [25]. H. Asmuni, E. K. Burke, and J. M. Garibaldi, "Fuzzy multiple heuristic ordering for course timetabling," in *Proc. 5th UK Workshop Comput. Intell.*, 2005, pp. 302–309.
- [26]. Y. Bykov, "Algorithm description," the 2002 International Timetabling Competition (TTC 2002), 2003.

- [27]. Hamed Babaeia, Jaber Karimpourb, Amin Hadidic, “A survey of approaches for university course timetabling problem”, *Computers & Industrial Engineering*, Volume 86, August 2015, Pages 43–59.
- [28]. R. Qu, E.K. Burke, B. McCollum, L.T.G. Merlot, A survey of search methodologies and automated system development for examination timetabling, *Journal of Scheduling* 12 (2009) 55–89.
- [29]. R. Lewis, A survey of metaheuristic-based techniques for university timetabling problems, *OR Spectrum* 30 (1) (2008) 167–190.
- [30]. Carter MW, Laporte G, Lee SY Examination timetabling: algorithmic strategies and applications. *Journal of the Operational Research Society* (1996) 74(3):373-383.
- [31]. Arani T, Lofti JA (1989) A three phased approach to final exam scheduling. *IIE Transactions* 21(4):86-96.
- [32]. Birbil, S.I., Fang, S.C.: An electromagnetism-like mechanism for global optimization. *J. Glob. Optim.* 25, 263–282 (2003)
- [33]. Dueck, G.: New Optimization Heuristics. The great deluge algorithm and the record-to-record travel. *J. Comput. Phys.* 104, 86–92 (1993)
- [34]. Abdullah, S., Turabieh, H.: Generating university course timetable using genetic algorithm and local search. In: *Proceeding of the 3rd International Conference on Hybrid Information Technology*, pp. 254–260 (2008)
- [35]. Abdullah, S., Burke, E.K., McCollum, B.: An investigation of variable neighborhood search for university course timetabling. In: *The 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA)*, pp. 413–427 (2005)
- [36]. McMullan, P.: An extended implementation of the great deluge algorithm for course timetabling. In: *Computational Science—ICCS, Part I. LNCS*, vol. 4487, pp. 538–545. Springer, Berlin (2007)
- [37]. Doerner K.F., Gendreau M., Greistorfer P., Gutjahr W. G., Hartl R.F., Reimann M., *Metaheuristics Progress in Complex Systems Optimization*, Springer Science+Business Media LLC, (2007), United State - New York.
- [38]. S.N.Sivanandam, S.N.Deepa. *Introduction to Genetic Algorithms*. Springer ebooks, Alemania (2008).
- [39]. Glover F., Kochenberger G.A., *Handbook of Metaheuristics*, Kluwer Academic Publishers (2002), United State - Massachusetts.
- [40]. Pardalos P.M., Romeijn H.E., *Handbook of Global Optimization*, Kluwer Academic Publishers (2002), Netherlands - Amsterdam
- [41]. Dirección Nacional de Censos y Encuestas, II CENSO NACIONAL UNIVERSITARIO 2010, INEI y ANR, (13/10/12), <http://blog.pucp.edu.pe/blog/wp-content/uploads/sites/452/2011/02/II-CNU-2010.pdf>
- [42]. Dirección de Estadística de ANR, Tips de Carreras Profesionales y Post Grado, Asamblea Nacional de Rectores, (22/10/12),

http://200.48.39.40/index.php?option=com_content&view=article&id=374&Itemid=724

- [43]. CONEAU, GUÍA PARA LA ACREDITACIÓN DE CARRERAS PROFESIONALES UNIVERSITARIAS DEL CONEAU, Ministerio de Educación, (11/10/12), <http://www.unmsm.edu.pe/occaa/documentos/Guia-parala%20Acreditacion-de-Carreras-Universitarias.pdf>
- [44]. Oficina de Imagen Institucional de la ANR, Directorio Universitario 2012, Asamblea Nacional de Rectores, (23/10/12), http://www.google.com.pe/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCAQFjAA&url=http%3A%2F%2Fwww.anr.edu.pe%2Findex9.php%3Foption%3Dcom_docman%26task%3Ddoc_download%26gid%3D1733&ei=TehUMetGoms9ATh3IDIDA&usg=AFQjCNEQEicmroeHeM1eRQIB_OUdDuenZw&sig2=mFbe5oENqFaP_xen8ugTgg
- [45]. <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf>
- [46]. Fundamentos de RUP, https://era.nih.gov/docs/rup_fundamentals.htm

ANEXO 1

ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

ÍNDICE

1. Introducción.....	89
1.1 Propósito.....	89
1.2 Visión General del ERS.....	89
2. Descripción General.....	89
2.1 Modelo de Casos de Uso.....	89
2.1.1 Casos de Uso.....	89
2.2 Especificación de Casos de Uso.....	89
2.2.1 Módulo de Configuración de Horarios.....	90
2.2.2 Módulo de Seguridad.....	107
2.2.3 Módulo de Elaboración de Horarios.....	120

1. Introducción

Este documento es una Especificación de Requisitos de Software (ERS) del sistema generador de horarios para la FISL. Esta especificación se ha realizado de acuerdo al estándar IEEE Recommended Practice for Software Requirements Specification IEEE Std 830-1998.

1.1. Propósito

El objetivo de la especificación es definir de manera clara y precisa las funcionalidades y restricciones del sistema que se desea construir.

1.2. Visión General del ERS

Este documento consta de 2 secciones. Esta sección es la Introducción y proporciona una visión general del ERS. En la Sección 2, se da una descripción general de la aplicación, con el fin de conocer las principales funciones que debe realizar.

2. Descripción General

En esta sección se describe a un alto nivel los Módulos de la aplicación a desarrollar. Se presentará el modelo de casos de uso (modelo que muestra la funcionalidad del módulo), las características de los usuarios, las suposiciones y dependencias del sistema.

2.1 Modelo de Casos de Uso

En la sección 5.1.1 del documento de la tesis se presenta el diagrama de paquetes, el diagramas de perfiles de usuario, los diagramas de casos de uso y los prototipos de cada caso de uso agrupados por los Módulo del sistema obtenidos durante el proceso de especificación de requisitos, los cuales permiten mostrar a un alto nivel las funcionalidades que la aplicación tendrá.

2.2 Especificación de Casos de Uso

A través de los casos de uso se consigue mostrar un alto nivel de la funcionalidad que tendrá el sistema de horarios a desarrollarse.

2.2.1 Módulo de Configuración de Horarios

Id	CU-CFH-001	Versión	01.00
Nombre	Gestionar Ciclo		
Actor Primario	Director de Escuela		
Descripción	Realizar el registro de un nuevo ciclo para la Facultad del Director de Escuela en sesión y la edición de datos y de la vigencia del ciclo.		
Pre-Condiciones	El usuario en sesión tiene el perfil de Director de Escuela o Administrador.		
Escenarios	Escenario Normal		
	Acciones del Actor	Respuesta del Sistema	
	1. El caso de uso empieza cuando el usuario selecciona la opción “Ciclo” del módulo de Configuración de Horarios.		
	2.	Muestra la interfaz “Ciclos” con la lista de ciclos asociados a la Facultad del usuario en forma descendente con los siguientes campos: <ul style="list-style-type: none"> • Nombre. • Fecha Inicial. • Fecha Final. • Estado. • Editar (Solo para estado 	

		<p>“Activo” o “Por Iniciar”)</p> <p>Además la opción “Registrar”.</p> <p>Nota de Implementación 1</p>
	<p>3. Selecciona la opción “Registrar”.</p> <p>Ver Escenario Alternativo 1</p>	
	4.	<p>Muestra la pantalla “Registro de Ciclo” con los campos:</p> <ul style="list-style-type: none"> • Nombre. • Fecha Inicial. • Fecha Final. • Estado. <p>Además la opción “Grabar”.</p> <p>Nota de Implementación 1, 2</p>
	5. Ingresar los campos solicitados.	
	6. Pulsa el botón “Grabar”.	
	7.	Registra el ciclo y actualiza la lista de ciclos con el nuevo registro.
	8.	Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
	1.1 Selecciona la opción “Editar” de un ciclo.	

		<p>Muestra la pantalla “Modificar Ciclo” con los campos:</p> <ul style="list-style-type: none"> • Nombre. • Fecha Inicial. • Fecha Final. • Estado (“Terminado”). <p>Además la opción “Grabar”.</p>
	Ingresa los datos a modificar.	
		Actualiza los datos modificados del ciclo y la lista de ciclos con el ciclo editado.
		Finaliza el proceso.
Post-Condiciones	Mostrar la pantalla de ciclos para una Facultad y permita la edición y registro de ciclos.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación	<ol style="list-style-type: none"> 1. El formato para las fechas es dd/mm/yyyy. 2. Si no existe en sesión un ciclo vigente para la Facultad del usuario entonces en la lista de estados muestra “Activo” y “Por iniciar” en el caso que exista solo muestra “Por iniciar”. 	

Id	CU-CFH-002	Versión	01.00
Nombre	Mantener Dictado de Clase		

Actor Primario	Director de Escuela	
Descripción	Registrar dictados de clases que se abrirán para un ciclo vigente.	
Pre-Condiciones	El usuario en sesión tiene el perfil de Director de Escuela o Administrador.	
Escenarios	Escenario Normal	
	Acciones del Actor (Usuario)	Respuesta del Sistema
	1. El caso de uso empieza cuando el usuario selecciona la opción “Dictado de Clase” del módulo de Configuración de Horarios.	
	2.	<p>Valida que exista un ciclo vigente en sesión.</p> <p>Ver Escenario Alternativo 1.</p>
	3.	<p>Muestra la interfaz “Dictados de Clases” con la lista de dictados de clases asociados al ciclo vigente con el filtro de ciclo académico y estos agrupados por asignatura. Los campos por cada dictado de clase son los siguientes:</p> <ul style="list-style-type: none"> • Grupo. • Modalidad. • Docente. <p>Opciones:</p> <ul style="list-style-type: none"> • Editar. • Eliminar. • Fijar. • Carga Lectiva. <p>Además la opción “Registrar”.</p> <p>✓ Si el usuario tiene el perfil</p>

		<p><i>Administrador</i> entonces mostrará el filtro de Dirección de Escuela.</p> <p>✓ Si el usuario tiene el perfil <i>Dirección de Escuela</i> entonces mostrará los Dictados de Clases registrados para esa Dirección de Escuela.</p> <p>Ver Nota de Implementación 1, 2.</p>
	<p>4. Selecciona la opción “Registrar”.</p> <p>Ver Escenario Alternativo 2.</p>	
	5.	<p>Muestra la interfaz “Registro de Dictado de Clase” con los siguientes campos:</p> <ul style="list-style-type: none"> • Opción seleccionada “Por N° de Ciclo”. • Opción “Por Asignatura”. • Lista de Facultades. • Lista de Dir. De Escuelas. • Lista de Planes de Estudios. • Lista de Ciclos Académicos. • Cantidad de Grupos. <p>Además la opción “Guardar”.</p> <p>Ver Nota de Implementación 2.</p>
	6. Ingresa los campos solicitados.	
	7. Pulsa el botón “Guardar”.	

	8.	Registra el Dictado de Clase y actualiza la lista de Dictados de Clases con el nuevo registro.
	9.	Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
	1.1	Valida que no exista un ciclo vigente en sesión entonces muestra mensaje “No hay ciclo vigente para mostrar dictados de clases registrados”.
		Finaliza el proceso.
	2.1 Selecciona la opción “Editar”.	
		Muestra la interfaz “Edición de Dictado de Clase” con los siguientes campos: <ul style="list-style-type: none"> • Asignatura (no editable). • Lista de Grupos. • Lista de Modalidades. • Lista de Docentes. Además la opción “Guardar”.
	Ingresa los datos a modificar.	
	Selecciona el botón “Guardar”.	

		Actualiza los datos modificados del dictado de clase y la lista de dictados de clases con el registro editado.
		Finaliza el proceso.
	2.2 Selecciona la opción “Eliminar”.	
		<p>Muestra la interfaz “Eliminación de Dictado de Clase” con el siguiente mensaje:</p> <p>¿Está seguro que desea eliminar el Dictado de Clase?</p> <ul style="list-style-type: none"> ✓ Asignatura. ✓ Grupo. ✓ Modalidad. ✓ Docente. <p>Además las opciones “Sí” y “No”.</p>
	Selecciona el botón “Sí”.	
		Elimina el dictado de clase y de la lista de dictados de clases.
		Finaliza el proceso.
	2.3 Selecciona la opción “Fijar”.	
		Muestra la interfaz “Fijar Dictado de Clase” con los siguientes campos:

		<p>Asignatura – Grupo – Modalidad – Docente.</p> <p>Tabla de hora vs día con el dictado de clase asignado en periodos fijos.</p> <p>✓ En caso se haya asignado un docente se debe deshabilitar las celdas que no estén dentro de su disponibilidad.</p> <p>Además la opción “Guardar”.</p>
	Modifica los periodos fijados asignado al dictado de clase.	
	Pulsa el botón “Guardar”.	
		Actualiza el horario asignado al dictado de clase.
		Finaliza el proceso.
	2.4 Selecciona la opción “Carga Lectiva”.	
		<p>Muestra la interfaz “Carga Lectiva” del Docente con los siguientes campos:</p> <ul style="list-style-type: none"> • Asignatura. • Grupo. • Modalidad. • Horas. <p>Además el total de horas y la</p>

		opción de “Cancelar”.
	Pulsa el botón “Cancelar”.	
		Cierra la interfaz “Carga Lectiva” y muestra la pantalla principal de Dictados de Clases.
		Finaliza el proceso.
Post-Condiciones	Mostrar la pantalla de dictados de clases para un ciclo vigente y permita la edición, la eliminación, la fijación del horario para el dictado de clase y visualizar la carga lectiva asociado al docente asignado.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación	<ol style="list-style-type: none"> 1. Si selecciona otro ciclo académico debe actualizar los dictados de clases agrupados por asignatura. 2. Si selecciona una Dirección de Escuela debe mostrar los Dictados de Clase registrados para esa Dirección de Escuela. 3. Si selecciona la opción “Por Asignatura” se muestra los campos: “Asignatura”. “Tipo de Dictado”. 	

Id	CU-CFH-003	Versión	01.00
Nombre	Configurar Periodos de Dictado por Ciclo		
Actor Primario	Director de Escuela		
Descripción	Actualizar los periodos en que se preferiría que se asignen los		

	dictados de clases según su ciclo.	
Pre-Condiciones	El usuario en sesión tiene el perfil de Director de Escuela o Administrador.	
Escenarios	Escenario Normal	
	Acciones del Actor (Usuario)	Respuesta del Sistema
	1. El caso de uso empieza cuando el usuario selecciona la opción “Conf. Periodo de Dictado” del módulo de Configuración de Horarios.	
	2.	<p>Muestra la interfaz “Periodos de Dictados de Asignaturas por Ciclos” que tiene los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Por cada ciclo académico el rango de horas por día que deberían asignarse los dictados de clase. ✓ Además muestra la opción Guardar.
	3. Modifica los periodos asignados para cada ciclo académico.	
	4. Selecciona la opción “Guardar”.	
	5.	Actualiza los periodos de dictado deseables para que se asignen a las

		asignaturas.
		Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
Post-Condiciones	Mostrar la pantalla de “Periodos de Dictados de Asignaturas por Ciclos” con el rango de horas cargadas que han sido asignadas.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación	N/A.	

Id	CU-CFH-004	Versión	01.00
Nombre	Configurar parámetros del algoritmo		
Actor Primario	Administrador		
Descripción	Actualizar los parámetros del algoritmo requeridos para generar el horario.		
Pre-Condiciones	El usuario en sesión tiene el perfil de Administrador.		
Escenarios	Escenario Normal		
	Acciones del Actor (Usuario)	Respuesta del Sistema	
	1. El caso de uso empieza cuando el usuario selecciona la opción “Configurar		

	Parámetros del Algoritmo” del módulo de Configuración de Horarios.	
	2.	<p>Muestra la interfaz “Configuración de Parámetros del Algoritmo” que tiene los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Tamaño de la Población. ✓ Cantidad de Iteraciones. ✓ Pc (Probabilidad de cruce). ✓ Pm (Probabilidad de mutación). ✓ Máximo de horas consecutivas por día. ✓ Máximo de horas totales por día. <p>Opción “Guardar”.</p>
	3. Ingresa los datos a modificar.	
	4. Pulsa el botón “Guardar”.	
	5.	Actualiza los datos de configuración de parámetros para el algoritmo.
	1.	Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
Post-Condiciones	Mostrar la pantalla “Configuración de Parámetros del	

	Algoritmo”.
Reglas de Negocio	N/A.
Requerimientos	N/A.
Notas de Implementación	N/A.

Id	CU-CFH-005	Versión	01.00
Nombre	Gestionar Disponibilidad Horaria		
Actor Primario	Docente		
Descripción	Actualizar la Disponibilidad del Docente para el ciclo vigente.		
Pre-Condiciones	El usuario en sesión tiene el perfil de Docente o Administrador.		
Escenarios	Escenario Normal		
	Acciones del Actor (Usuario)	Respuesta del Sistema	
	1. El caso de uso empieza cuando el usuario selecciona la opción “Disponibilidad Docente” del módulo de Configuración de Horarios.		
	2.	Muestra la interfaz “Disponibilidad Docente” que tiene los siguientes datos: ✓ Nombre del Docente. ✓ Tabla hora por día con los	

		periodos electos por el docente como su disponibilidad. ✓ Opción de “Guardar”.
	3. Ingresar periodos de disponibilidad.	
	4. Selecciona la opción “Guardar”.	
	5.	Actualiza la Disponibilidad Horaria del Docente.
	6.	Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
Post-Condiciones	Mostrar la pantalla de la disponibilidad horaria del Docente.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación	N/A.	

Id	CU-CFH-006	Versión	01.00
Nombre	Gestionar Clases de preferencias		
Actor Primario	Docente		
Descripción	Registrar las clases de preferencia para el Docente.		

Pre-Condiciones	El usuario en sesión tiene el perfil de Docente o Administrador.	
Escenarios	Escenario Normal	
	Acciones del Actor (Usuario)	Respuesta del Sistema
	1. El caso de uso empieza cuando el usuario selecciona la opción “Clases de Preferencia” del módulo de Configuración de Horarios.	
	2.	Muestra la interfaz “Clases de Preferencia” con la lista de clases con los siguientes campos: ✓ Curso (Asignatura y Modalidad de Dictado). ✓ Dirección de Escuela. Además la opción de “Registrar”.
	3. Selecciona la opción “Registrar”.	
	4.	Muestra la lista de Clases agrupados por Ciclo Académico (Nombre de la Asignatura – Modalidad de Dictado). Se encuentran seleccionadas las clases de preferencia. Además la opción de “Registrar”.
	5. Selecciona las clases de su preferencia.	

	6. Selecciona la opción de “Registrar”.	
	7.	Actualiza las clases de preferencia del Docente.
	8.	Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
Post-Condiciones	Mostrar la pantalla de Clases de preferencia del Docente.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación	N/A.	

Id	CU-CFH-007	Versión	01.00
Nombre	Asignar Dictado de Clase		
Actor Primario	Jefe de Departamento		
Descripción	Registrar dictados de clases que se abrirán para un ciclo vigente.		
Pre-Condiciones	El usuario en sesión tiene el perfil de Director de Escuela o Administrador.		
Escenarios	Escenario Normal		
	Acciones del Actor (Usuario)	Respuesta del Sistema	

	1. El caso de uso empieza cuando el usuario selecciona la opción “Asignar Dictado de Clase” del módulo de Configuración de Horarios.	
	2.	Valida que exista un ciclo vigente en sesión. Ver Escenario Alternativo 1.
	3.	Muestra la interfaz “Asignar Dictado de Clase” con los siguientes datos: ✓ La lista de dictados de clases agrupadas por ciclo y asignatura del ciclo vigente. ✓ Lista de Docentes con la lista de Clases de su preferencia.
	4. Selecciona el Docente.	
	5.	Se muestra los Dictados de Clase que tiene asignado. Se muestra las Clases de su preferencia en modo informativo para el Jefe de Departamento.
	6. Selecciona el Dictado de Clase a asignar.	
	7. Arrastra el Dictado de Clase hacia la lista de Dictados de Clases asignados al Docente.	

	8.	Se actualiza la lista de Dictados de Clases del Docente.
	9.	Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
	1.1	Valida que no exista un ciclo vigente en sesión entonces muestra mensaje “No hay ciclo vigente para mostrar dictados de clases registrados”.
		Finaliza el proceso.
Post-Condiciones	Mostrar la pantalla de Asignación de Dictados de Clases del Docente para el ciclo vigente.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación	N/A.	

2.2.2 Módulo de Seguridad

Id	CU-SEG-001	Versión	01.00
Nombre	Loguear		
Actor Primario	Usuario		

Descripción	Realizar la validación del Usuario para poder acceder al Sistema de Programación de Horarios y muestra la lista de opciones que tenga asociado el perfil por defecto del Usuario.	
Pre-Condiciones	N/A.	
Escenarios	Escenario Normal	
	Acciones del Actor	Respuesta del Sistema
	1. El caso de uso empieza cuando ingresa a la interfaz “Ingreso al Sistema”.	
	2. Ingresa el usuario y la contraseña.	
	3. El usuario selecciona el botón “Aceptar”.	
	4.	Valida que el <i>Usuario</i> sea diferente a “”. (Ver Escenario Alternativo 2).
	5.	Valida que la <i>Contraseña</i> sea diferente a “”. (Ver Escenario Alternativo 3).
	6.	Consulta los siguientes datos obtenidos con el usuario ingresado: <ul style="list-style-type: none"> ▪ Nombres. ▪ Apellidos. ▪ Usuario. ▪ Password. ▪ Género. ▪ Email. ▪ Facultad. ▪ Lista de Perfiles asociados:

		<ul style="list-style-type: none"> ▪ Código. ▪ Descripción. ▪ Lista de Opciones de Menú: <ul style="list-style-type: none"> ❖ Código. ❖ Código del Padre. ❖ Nombre. ❖ Url. ❖ Nivel. ❖ Orden. ❖ Estado. ❖ Icono.
	7.	<p>Valida que el <i>Usuario</i> retornado sea diferente a vacío.</p> <p>(Ver Escenario Alternativo 5).</p>
	8.	<p>Valida que la <i>Contraseña del Usuario</i> retornado sea igual a la <i>Contraseña</i> ingresada.</p> <p>(Ver Escenario Alternativo 6).</p>
	9.	<p>Valida que el <i>Usuario</i> tenga al menos un perfil asociado.</p> <p>(Ver Escenario Alternativo 7).</p>
	10.	<p>Verifica que existe ciclo vigente (estado igual a 1) asociado a la Facultad del usuario.</p> <p>(Ver Escenario Alternativo 8).</p>
	11.	<p>Muestra el nombre del usuario, el perfil y el ciclo vigente en la cabecera de la página de inicio.</p>
	12.	<p>Muestra la pantalla principal con la Lista de opciones asociados al Perfil por</p>

		defecto del usuario. (Ver Nota de Implementación 1).
	13.	Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
	1.1	Valida que el <i>Usuario</i> sea diferente a "" entonces muestra mensaje: "Ingrese Usuario."
		Finaliza el proceso.
	3.1	Valida que la <i>Contraseña</i> sea diferente a "" entonces muestra mensaje: "Ingrese Contraseña."
		Finaliza el proceso.
	5.1	Valida que el <i>Usuario</i> retornado sea igual a vacío entonces muestra mensaje: "No se encuentra el Usuario registrado."
		Finaliza el proceso.
	6.1	Valida que la <i>Contraseña del Usuario</i> retornado sea diferente a la <i>Contraseña</i> ingresada entonces muestra mensaje: "La Contraseña es incorrecta."
		Finaliza el proceso.
	7.1	Valida que el <i>Usuario</i> no tiene perfil

		asociado entonces muestra mensaje: “Usuario no tiene Perfil asociado.”
		Finaliza el proceso.
	8.1	Verifica que no existe ciclo vigente asociado a la Facultad del usuario.
		Muestra el nombre del usuario, el perfil y que no hay ciclo vigente en la cabecera de la página de inicio.
		Retorna al paso 12.
Post-Condiciones	Mostrar la pantalla de Inicio con las opciones habilitadas del Menú que tenga asociado el perfil por defecto del Usuario.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación	<p>1. Según el perfil del usuario muestra la lista de opciones asociada:</p> <p>✓ Perfil es igual a <i>Administrador</i>.</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Dirección de Escuela. - Plan de Estudios. - Asignaturas. - Docentes. - Clases. - Salones. - Grupos. <p><i>Configuración de Horarios:</i></p> <ul style="list-style-type: none"> - Ciclo. - Dictados de Clases. - Asignar Clases. - Disponibilidad Docente. 	

	<ul style="list-style-type: none"> - Conf. Periodo de Dictado. - Configurar Parámetros. - Clases de Preferencia. <p><i>Elaboración de Horarios:</i></p> <ul style="list-style-type: none"> - Gestionar Horario. - Visualizar Horario. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Perfiles. - Usuarios. - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Perfil es igual a <i>Docente</i>.</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Edición Docente. <p><i>Configuración de Horarios:</i></p> <ul style="list-style-type: none"> - Disponibilidad Docente. - Clases de Preferencia. <p><i>Elaboración de Horarios:</i></p> <ul style="list-style-type: none"> - Visualizar Horario. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Perfil es igual a <i>Director de Escuela</i>.</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Dirección de Escuela.
--	---

	<ul style="list-style-type: none"> - Plan de Estudios. - Asignaturas. - Clases. - Grupos. <p><i>Configuración de Horarios:</i></p> <ul style="list-style-type: none"> - Ciclo. - Dictados de Clases. - Conf. Periodo de Dictado. <p><i>Elaboración de Horarios:</i></p> <ul style="list-style-type: none"> - Gestionar Horario. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Perfil es igual a <i>Jefe de Departamento</i>.</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Dirección de Escuela. - Plan de Estudios. - Asignaturas. - Docentes. <p><i>Configuración de Horarios:</i></p> <ul style="list-style-type: none"> - Asignar Clases. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Perfil es igual a <i>Asistente de Logística</i>.</p>
--	--

	<p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Salones. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Perfil es igual a <i>Asistente de Laboratorio</i>.</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Laboratorios. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil.
--	--

Id	CU-SEG-002	Versión	01.00
Nombre	Cambiar Perfil		
Actor Primario	Usuario		
Descripción	Realizar el cambio de perfil para actualizar la lista de opciones que tenga asociado el perfil seleccionado.		
Pre-Condiciones	Se debe tener un perfil por defecto, los datos del Usuario y la lista de opciones del Menú.		
Escenarios	Escenario Normal		
	Acciones del Actor	Respuesta del Sistema	

	(Usuario)	
	9. El caso de uso empieza cuando el usuario selecciona la opción “Cambiar Perfil” del módulo de Seguridad.	
	10.	Muestra la interfaz “Cambio de Perfil” con la lista de Perfiles asociados al Usuario y la opción de asignar <i>Perfil por defecto</i> .
	11. Selecciona un <i>Perfil</i> diferente al <i>Perfil mostrado</i> .	
	12. Selecciona la opción de asignar <i>Perfil por defecto</i> .	
	13. Pulsa el botón <i>Actualizar</i> .	
	14.	Valida que la opción de asignar <i>Perfil por defecto</i> este chequeada. (Ver Escenario Alternativo 6).
	15.	Guarda la opción por defecto al Usuario con los siguientes parámetros: <ul style="list-style-type: none"> • Código de Usuario. • Código de Perfil seleccionado. • Estado (2).
	16.	✓ Valida que el perfil seleccionado sea igual a

		<p><i>Administrador.</i> Actualiza la lista de opciones:</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Dirección de Escuela. - Plan de Estudios. - Asignaturas. - Docentes. - Clases. - Salones. - Grupos. <p><i>Configuración de Horarios:</i></p> <ul style="list-style-type: none"> - Ciclo. - Dictados de Clases. - Asignar Clases. - Disponibilidad Docente. - Conf. Periodo de Dictado. - Configurar Parámetros. - Clases de Preferencia. <p><i>Elaboración de Horarios:</i></p> <ul style="list-style-type: none"> - Gestionar Horario. - Visualizar Horario. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Perfiles. - Usuarios. - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Valida que el perfil seleccionado sea igual a <i>Docente</i>. Actualiza la lista de opciones:</p>
--	--	--

		<p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Edición Docente. <p><i>Configuración de Horarios:</i></p> <ul style="list-style-type: none"> - Disponibilidad Docente. - Clases de Preferencia. <p><i>Elaboración de Horarios:</i></p> <ul style="list-style-type: none"> - Visualizar Horario. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Valida que el perfil seleccionado sea igual a <i>Director de Escuela</i>. Actualiza la lista de opciones:</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Dirección de Escuela. - Plan de Estudios. - Asignaturas. - Clases. - Grupos. <p><i>Configuración de Horarios:</i></p> <ul style="list-style-type: none"> - Ciclo. - Dictados de Clases. - Conf. Periodo de Dictado. <p><i>Elaboración de Horarios:</i></p> <ul style="list-style-type: none"> - Gestionar Horario.
--	--	--

		<p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Valida que el perfil seleccionado sea igual a <i>Jefe de Departamento</i>. Actualiza la lista de opciones:</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Dirección de Escuela. - Plan de Estudios. - Asignaturas. - Docentes. <p><i>Configuración de Horarios:</i></p> <ul style="list-style-type: none"> - Asignar Clases. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil. <p>✓ Valida que el perfil seleccionado sea igual a <i>Asistente de Logística</i>. Actualiza la lista de opciones:</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Salones. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña.
--	--	--

		<ul style="list-style-type: none"> - Cambiar Perfil. <p>✓ Valida que el perfil seleccionado sea igual a <i>Asistente de Laboratorio</i>. Actualiza la lista de opciones:</p> <p><i>Mantenedores:</i></p> <ul style="list-style-type: none"> - Laboratorios. <p><i>Seguridad:</i></p> <ul style="list-style-type: none"> - Cambiar Contraseña. - Cambiar Perfil.
	17.	Finaliza el proceso.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
	6.1	Valida que la opción de asignar <i>Perfil por defecto</i> no este chequeada.
		Retorna al paso 8.
Post-Condiciones	Mostrar la pantalla de Inicio con las opciones actualizadas del Menú que tenga asociado el perfil seleccionado.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación	N/A.	

2.2.3. Módulo de Elaboración de Horarios

Id	CU-ELH-001	Versión	01.00
Nombre:	Gestionar Horario		
Actor Primario	Director de escuela		
Descripción	Generar un horario con los datos previos de la configuración del horario, podrá consultar los horarios que se hayan generado y podrá generar los reportes: Salón por periodo, Horarios por salón y Horario por ciclo		
Pre-Condiciones	El usuario debe estar logueado.		
Escenarios	Escenario Normal		
	Acciones del Actor (Usuario)	Respuesta del Sistema	
	1. El caso de uso empieza cuando el usuario selecciona la opción “Gestionar Horario” del módulo de <i>Elaboración de Horarios</i> .		
	2.	Muestra la interfaz “Gestionar Horario” y verifica si existe un ciclo vigente.	
	3.	Muestra el ciclo vigente y los horarios registrados para ese ciclo ordenados de acuerdo a la función de aptitud del horario, la opción “Registrar” y una tabla con los siguientes campos: <ul style="list-style-type: none"> • Código. • Función de Aptitud. 	

		<p>Además la tabla muestra las opciones:</p> <ul style="list-style-type: none"> • Visualizar Horario. • Visualizar Parámetro. • Exportar Horario por Periodo. • Exportar Horario por Salón. • Exportar Horario por Ciclo. • Eliminar.
	4. Selecciona la opción “Registrar”.	
	5.	Añade un horario al ciclo vigente.
	6.	Los pasos 4 y 5 se repiten si el usuario desea registrar más horarios.
	7.	El CUS finaliza.
	Escenario Alternativo	
	Acciones del Actor	Acciones del Sistema
	6.2	
	8.1	

	8.2	
Post-Condiciones	N/A.	
Reglas de Negocio	N/A.	
Requerimientos	N/A.	
Notas de Implementación		

ANEXO 2

DOCUMENTO DE DISEÑO

ÍNDICE

1. Objetivo.....	123
2. Alcance.....	123
2.1 Referencias.....	123
2.2 Visión General del documento.....	123
3. Diseño de la Base de Datos.....	123
4. Diagrama de Secuencia.....	128
4.1 Módulo de Elaboración de Horarios.....	128
4.1.1 Gestionar Ciclo.....	129
4.1.2 Mantener Dictado de Clase.....	130
4.1.3 Configurar Periodos de Dictado por Ciclo.....	131
4.1.4 Configurar parámetros del algoritmo.....	131
4.1.5 Gestionar Disponibilidad Horaria.....	132
4.1.6 Gestionar Clases de preferencias.....	132
4.2 Módulo de Configuración de Horarios.....	133
4.2.1 Gestionar Horario.....	133

1. Objetivo

El objetivo del presente documento es mostrar a un nivel más bajo el funcionamiento del “Sistema de Generación de Horarios” realizado como proyecto de tesis.

Para ello hacemos uso de las herramientas de la técnica de modelamiento UML, como son los diagramas de secuencia y de las clases de diseño, para mostrar la composición lógica de la secuencia de codificación de cada módulo del sistema.

2. Alcance

El alcance de este documento incluye únicamente los casos de uso. Estos se explican con más detalle en el documento Especificación de Requisitos de Software.

2.1 Referencias

Las referencias utilizadas para el documento son:

- Documento de Tesis
- ERS

2.2 Visión general del documento

Este documento presentará de manera clara el diseño de las estructuras escogidas, para el soporte de los datos y procesos involucrados en el sistema, empleando un lenguaje sencillo y directo, con la ayuda de gráficos con el fin de dar una mejor visión de lo argumentado.

3. Diseño de la Base de Datos

Los datos del “Sistema de Generación de Horarios” se almacenarán en tablas alojadas en una base de datos, los cuales a su vez servirán para guardar información que utilizará el sistema. A continuación se mostrará el Diccionario de Datos de las principales tablas:

Universidad				
Campo	Tipo de Dato	PK	FK	Descripción
nUnvId	INT	SI	NO	Almacena el código de la Universidad.
vNombre	VARCHAR(60)	NO	NO	Almacena el nombre de la Universidad.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) la Universidad por defecto toma el valor 1.

Facultad				
Campo	Tipo de Dato	PK	FK	Descripción
nFctId	INT	SI	NO	Almacena el código de la Facultad.
nUnvId	INT	NO	SI	Almacena el código de la Universidad.
vNombre	VARCHAR(60)	NO	NO	Almacena el nombre de la Facultad.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) la Facultad por defecto toma el valor 1.

DireccioneDeEscuela				
Campo	Tipo de Dato	PK	FK	Descripción
nDrcEscId	INT	SI	NO	Almacena el código de la Dirección de Escuela.
nFctId	INT	NO	SI	Almacena el código de la Facultad.
vNombre	VARCHAR(60)	NO	NO	Almacena el nombre de la Dirección de Escuela.
dtFchIni	DATE	NO	NO	Almacena la fecha de Creación de la Dirección de Escuela.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) la Dirección de Escuela por defecto toma el valor 1.

Asignatura				
Campo	Tipo de Dato	PK	FK	Descripción
nAsgId	INT	SI	NO	Almacena el id de la Asignatura.
nPlnCrrId	INT	NO	SI	Almacena el código del Plan Curricular.
nCicAcId	INT	NO	SI	Almacena el código del Ciclo Académico.
vCodigo	VARCHAR(20)	NO	NO	Almacena el código de la Asignatura.
vNombre	VARCHAR(60)	NO	NO	Almacena el nombre de la Asignatura.
nCreditos	INT	NO	NO	Almacena el número de Créditos de la Asignatura.
vElectivo	VARCHAR(1)	NO	NO	Indica si es electivo (1) o no (0) la Asignatura por defecto toma el valor 0.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) la Asignatura por defecto toma el valor 1.

Clase				
Campo	Tipo de Dato	PK	FK	Descripción
nClsId	INT	SI	NO	Almacena el código la Clase.
nAsgId	INT	NO	SI	Almacena el código de la Asignatura.
nMdlDctId	INT	NO	NO	Almacena la modalidad de dictado: teórico (1), práctico (2) o laboratorio (3).
nHoras	DECIMAL(4,2)	NO	NO	Almacena el número de horas de dictado de la clase.
nCapacidad	INT	NO	NO	Almacena la capacidad de alumnos para la clase.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) la Clase por defecto toma el valor 1.

DocentexClase				
Campo	Tipo de Dato	PK	FK	Descripción
nDcnClsId	INT	SI	NO	Almacena el código la Clase.
nCicId	INT	NO	SI	Almacena el código de la Asignatura.
nClsId	INT	NO	SI	Almacena la modalidad de dictado: teórico (1), práctico (2) o laboratorio (3).
nHoras	INT	NO	NO	Almacena el número de horas de dictado de la clase.
nDcnId	INT	NO	NO	Almacena la capacidad de alumnos para la clase.
nGrpId	INT	NO	NO	Almacena el grupo al que pertenece la clase.
nOrden	INT	NO	NO	Almacena el orden (diferencia entre la disponibilidad del docente y su carga lectiva) en que se asignaran los dictados de clase para el algoritmo.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) la Clase por defecto toma el valor 1.

ClasexClase				
Campo	Tipo de Dato	PK	FK	Descripción
nDcnClsId1	INT	SI	NO	Almacena el código del primer dictado de clase que no que no puede cruzarte con el segundo dictado de clase.
nDcnClsId2	VARCHAR(60)	SI	NO	Almacena el código del segundo dictado de clase que no que no puede cruzarte con el primer dictado de clase.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) los dictados de clase conflictantes por defecto toma el valor 1.

DocentexPeriodo				
Campo	Tipo de Dato	PK	FK	Descripción
nDcnId	INT	SI	NO	Almacena el código del docente.
nPrdId	INT	SI	NO	Almacena el código del periodo.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) el periodo del docente por defecto toma el valor 1.

ClasexSalon				
Campo	Tipo de Dato	PK	FK	Descripción
nDcnClsId	INT	SI	NO	Almacena el código del dictado de dictado de clase.
nSlnId	INT	SI	NO	Almacena el código del salón.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) del salón apto para el dictado de clase por defecto toma el valor 1.

AsignaturaxPeriodo				
Campo	Tipo de Dato	PK	FK	Descripción
nAsgPrdId	INT	SI	NO	Almacena el id de la asignatura y periodo.
nAsgId	INT	NO	SI	Almacena el código de la asignatura.
nPrdId	INT	NO	SI	Almacena el código del periodo.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) el periodo de la asignatura por defecto toma el valor 1.

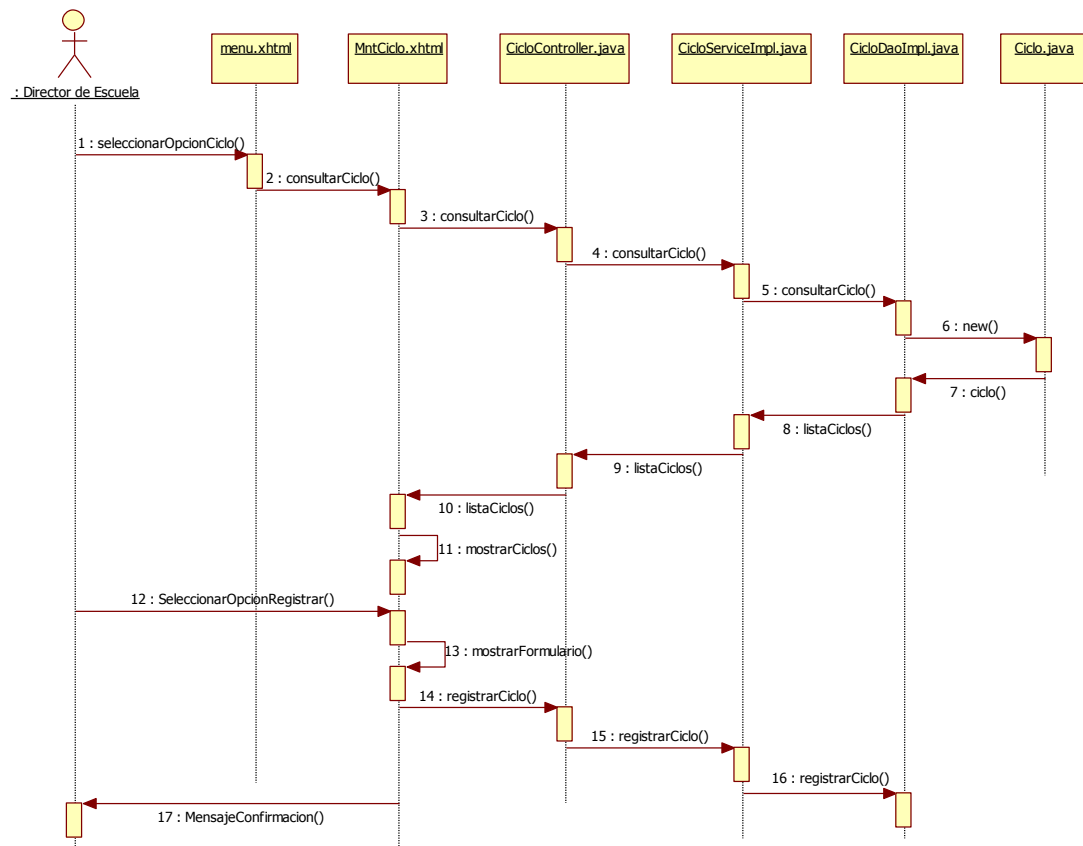
SalonxPeriodo				
Campo	Tipo de Dato	PK	FK	Descripción
nSlnId	INT	SI	NO	Almacena código del salón.
nPrdId	INT	NO	SI	Almacena el código del periodo.
vEstado	VARCHAR(1)	NO	NO	Almacena el estado e indica si está activo (1) o no (0) el periodo del salón por defecto toma el valor 1.

4. Diagrama de Secuencia

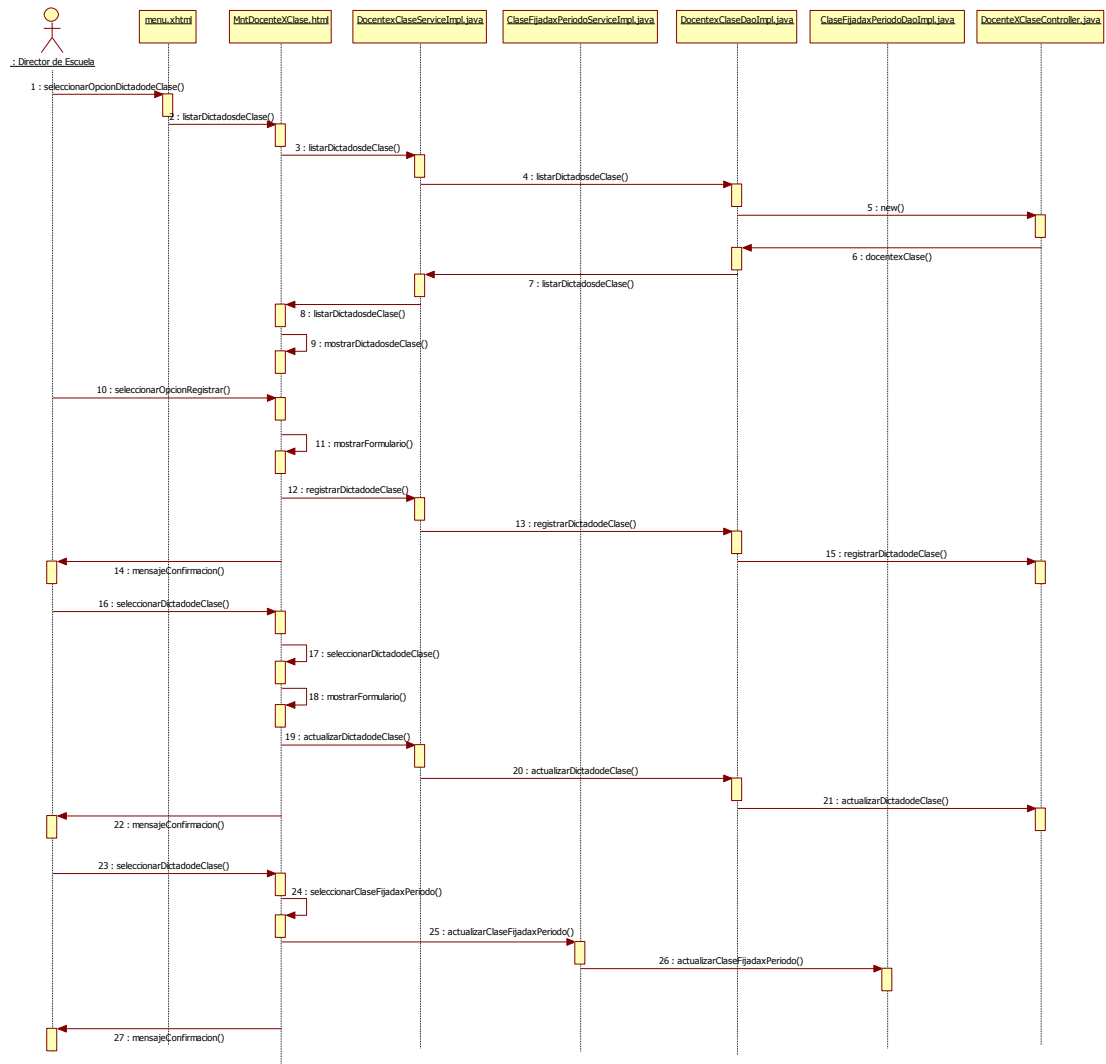
En los diagramas de secuencias se muestran a detalle los pasos por cada una de las capas, definidas en la arquitectura, para la realización de las funcionalidades descritas en el sistema.

4.1 Módulo de Configuración de Horarios

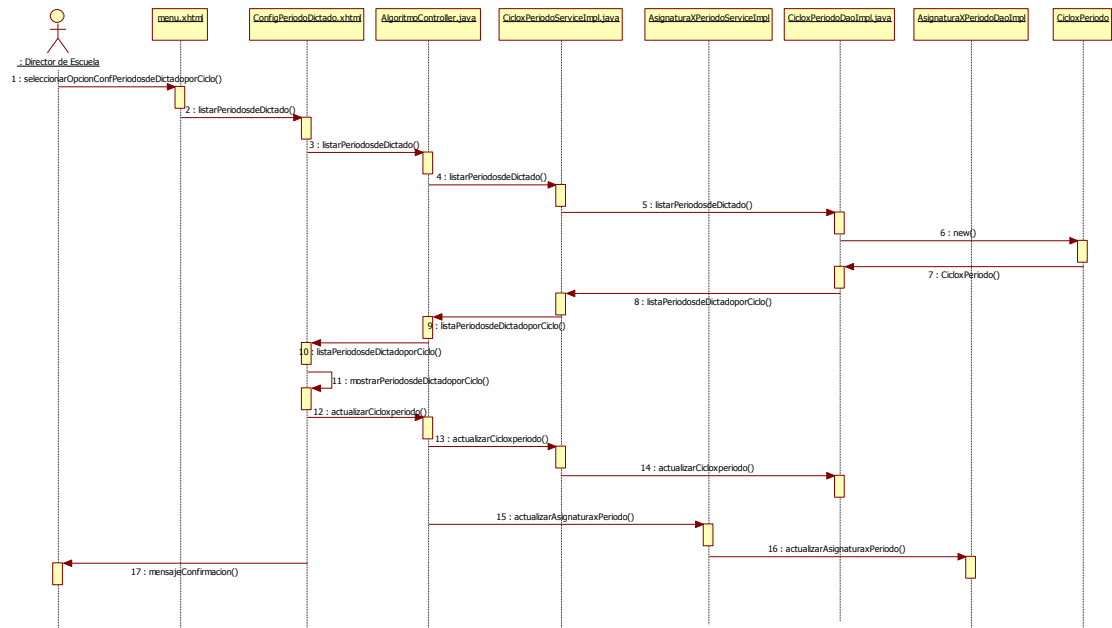
4.1.1 Gestionar Ciclo



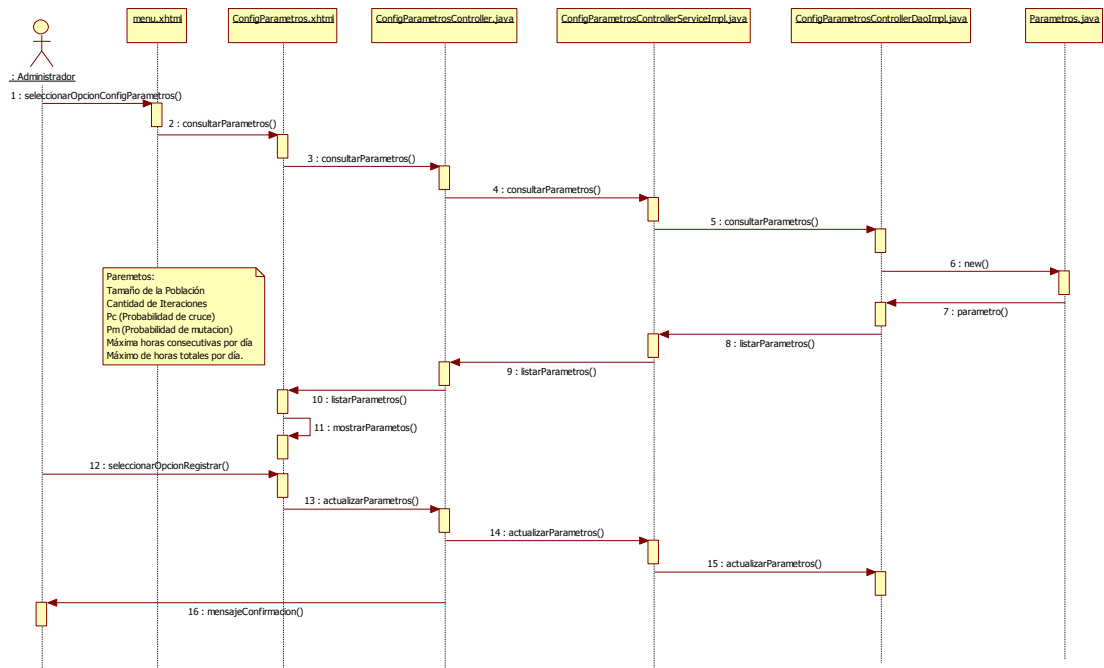
4.1.2 Mantener Dictado de Clase



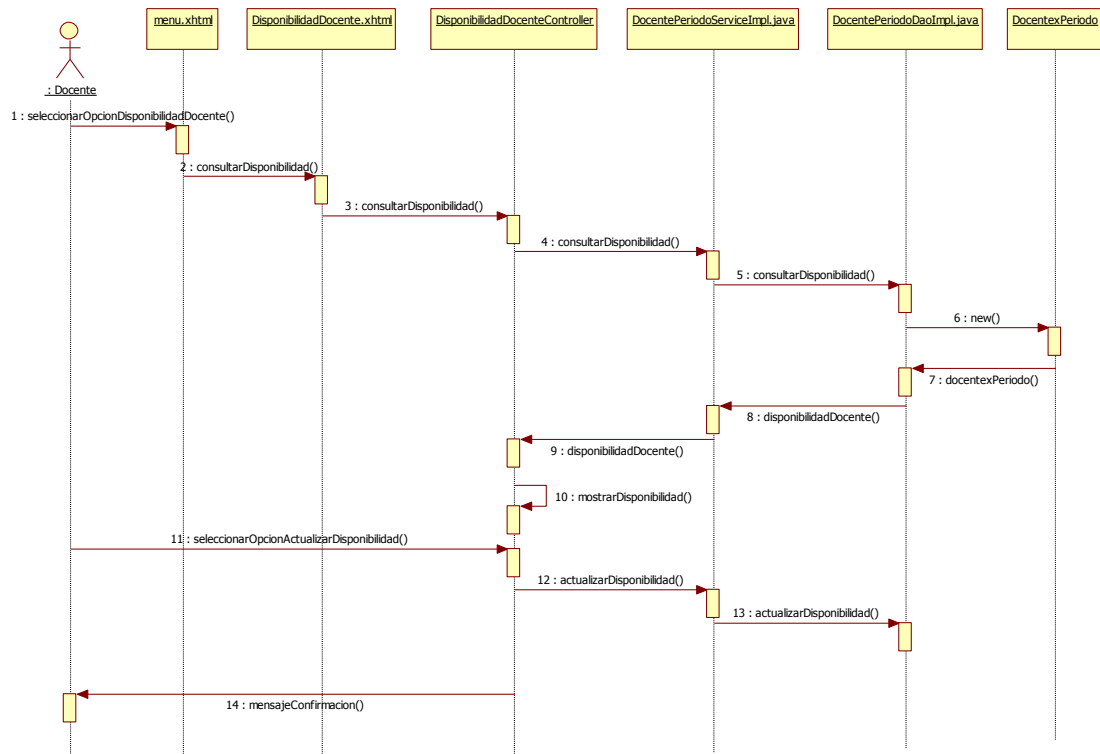
4.1.3 Configurar Periodos de Dictado por Ciclo



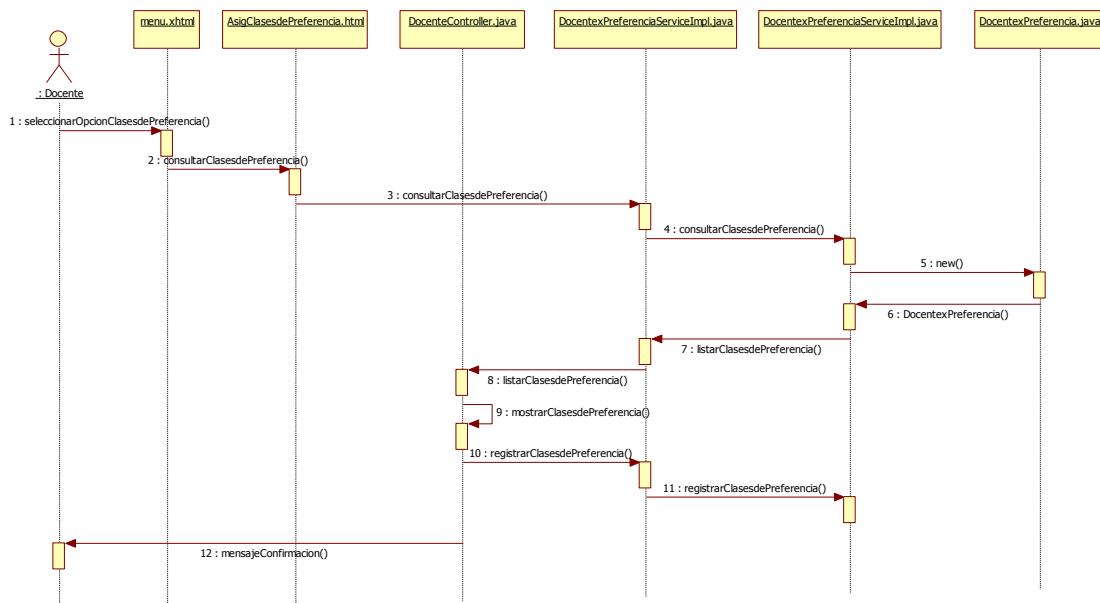
4.1.4 Configurar parámetros del algoritmo



4.1.5 Gestionar Disponibilidad Horaria

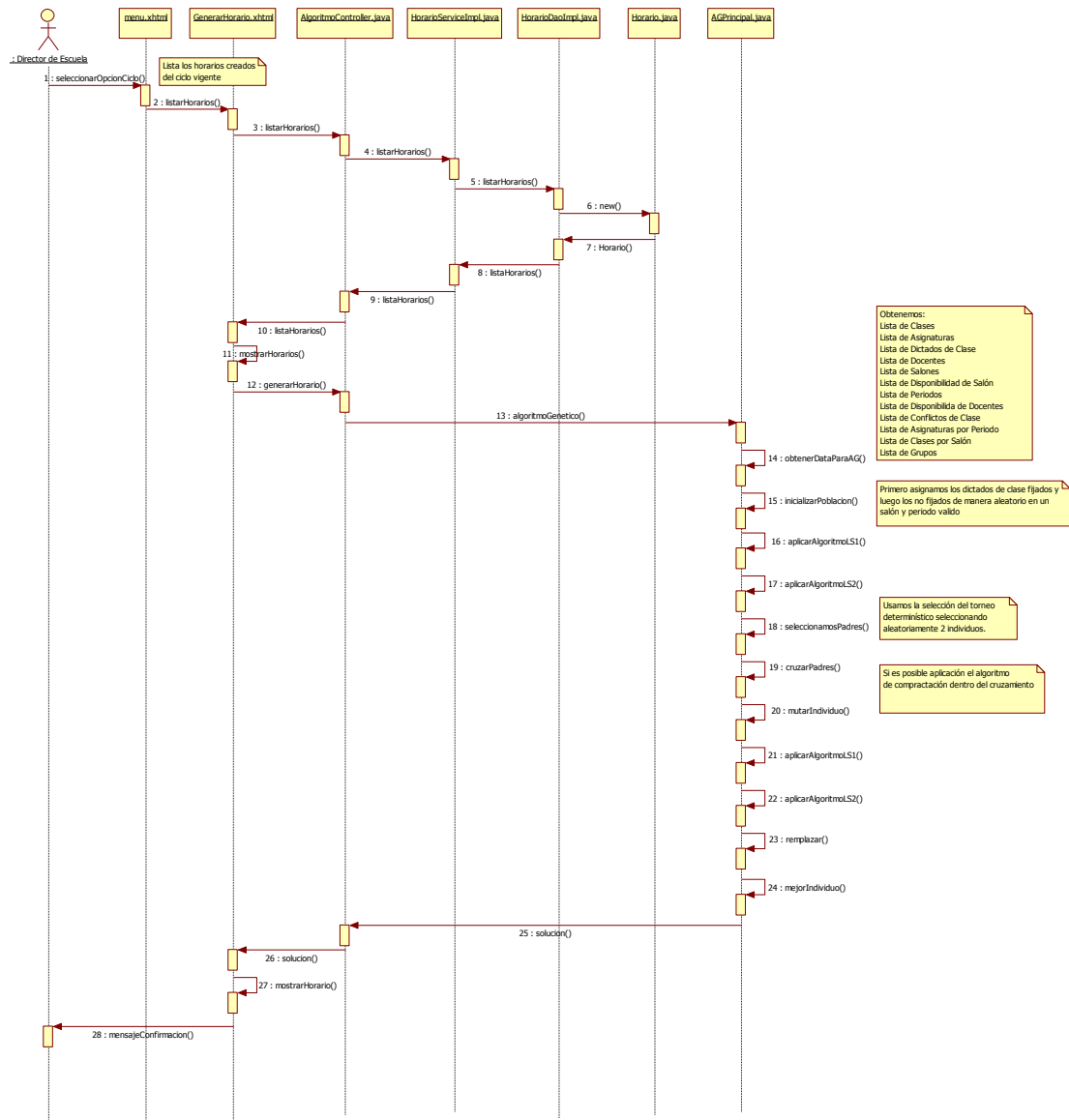


4.1.6 Gestionar Clases de preferencias



4.2 Módulo de Configuración de Horarios

4.2.1 Gestionar Horario



ANEXO 3

DOCUMENTO DE ARQUITECTURA

ÍNDICE

1. Introducción	134
2. Vista General de la Arquitectura.....	134
2.1. Capa de presentación	135
2.2. Capa de Lógica de Negocio.....	135
2.3. Capa de acceso a datos	135
3. Restricciones de la Arquitectura.....	136
4. Vista Lógica del Sistema	137
5. Vista de Implementación	138
6. Vista de Despliegue	139
7. Modelo de Datos.....	140
7.1. Modelo Físico	140

1. Introducción

Al realizar el diseño del software de Sistema de Generación de Horarios se identifica el plan detallado del sistema a nivel de componentes, relaciones entre sus elementos, reglas de negocio, todo ello con la finalidad de tener una guía de implementación que permita a futuro tener una aplicación con las características de ingeniería de software (mantenibilidad, confiabilidad, eficiencia y usabilidad).

Para desarrollar el software es necesario contar con una arquitectura definida, que permita representar la estructura del sistema, para ello este documento pretende dejar en claro la visión entre las personas involucradas en el análisis, desarrollo e implementación del sistema.

2. Vista General de la Arquitectura

El diseño del Sistema de Generación de Horarios usará el Patrón de Arquitectura Modelo Vista Controlador (MVC), detallamos la distribución en el siguiente cuadro:

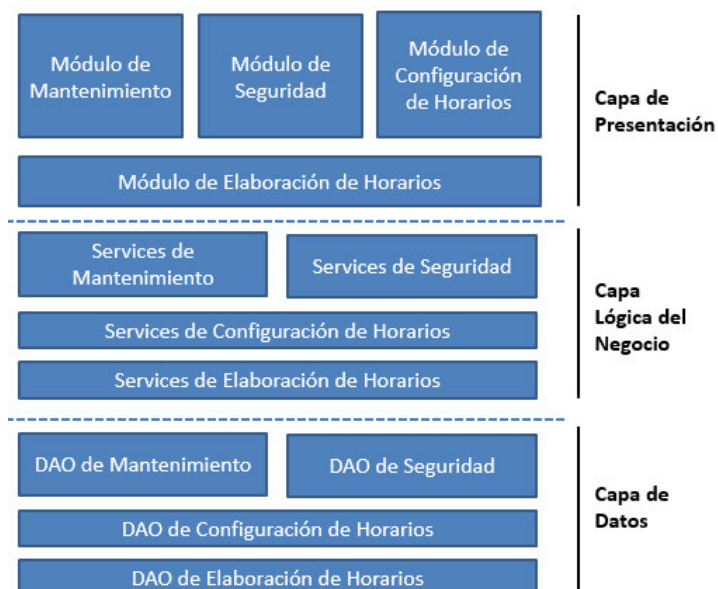


Figura 1 Modelo Vista Controlador del Sistema de Generación de Horarios [Fuente: Elaboración Propia].

El patrón de arquitectura (Modelo – Vista – Controlador), separa por capas los datos de la lógica del negocio y de la interfaces del sistema, la base de este patrón es la reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento, las tres capas en las que se divide la arquitectura son las siguientes:

2.1 Capa de presentación

También conocida como “Capa de Usuario”, es la capa donde se tendrá las interfaces que le permitan al usuario interactuar con el sistema en sus diferentes módulos, esta capa tiene comunicación con la capa de negocio a través del manejo de controladores para lo cual utilizaremos Framework: Primefaces 3.5.

El sistema de generación de horarios maneja en esta capa los siguientes módulos:

1. Módulo de Configuración de Horarios.
2. Módulo de Seguridad.
3. Módulo de Mantenimiento.
4. Módulo de Elaboración de Horarios

2.2 Capa de Lógica de Negocio

Esta capa provee todo lo que es la lógica del negocio, el Sistema de Generación de Horarios manejará servicios por cada módulo ya mencionado en la capa de presentación, y maneja servicios para la invocación de los DAO.

2.3 Capa de acceso a datos

Esta capa provee los servicios y conexiones a la base de datos requeridos por la capa de lógica de Negocio, se manejará por medio del patrón de Diseño DAO (Data Access Object) por cada módulo que tiene el sistema.

Por otro lado, el manejador de base de datos utilizado para este sistema será MySQL 5.1.

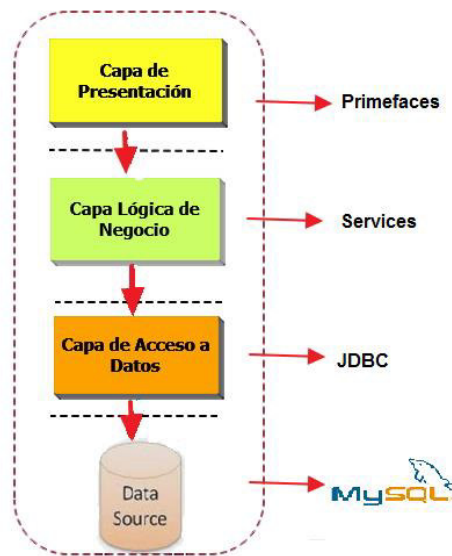


Figura 2 Patrón de Arquitectura MVC Sistema de Generación de Horarios [Fuente: Elaboración Propia].

3. Restricciones de la Arquitectura

Hemos identificado a lo largo del análisis y diseño de la arquitectura, ciertos requerimientos no funcionales que tendrán impacto sobre ella:

<i>RNF</i>	<i>Descripción del impacto sobre la arquitectura</i>
Requerimientos de Usabilidad	
RNF01	La interfaz debe ser amigable al usuario, siendo que el usuario no debería necesitar un entrenamiento especializado para el uso de ella misma.
RNF02	Los reportes deben estar diseñados para su correspondiente exportación en formato PDF y excel.
Requerimiento de Confiabilidad	
RNF03	El sistema debe estar operativo al 99% durante todo el proceso de configuración y elaboración de horarios para cada matrícula que se realice durante el año académico.

Requerimiento de Rendimiento	
RNF04	Tiempo de respuesta para la elaboración de horarios debe ser entre 10 y 30 minutos (según la cantidad de dictados de clase y recursos a asignar).
Requerimiento de Soporte	
RNF05	El software debe de ser compatible con estaciones de trabajo con procesador Mínimo Intel Core i5, con un mínimo de 3.86 GB de memoria RAM.
<i>Requerimiento de Diseño</i>	
RNF06	La solución se desarrollará usando Java 6 como lenguaje de programación.
RNF07	Se ha decidido diseñar el software con el Patrón de Arquitectura Modelo Vista Controlador (MVC).
RNF08	El Framework que se utilizará será Primefaces 3.5.
RNF09	El JDK utilizado es jdk1.6.25 o superior para el desarrollo del software.
RNF10	Como manejador de base de datos se utilizara MySQL 5.1.

4. Vista Lógica Del Sistema

Se ha desarrollado el diagrama de la Vista Lógica del Sistema de Generación de Horarios:

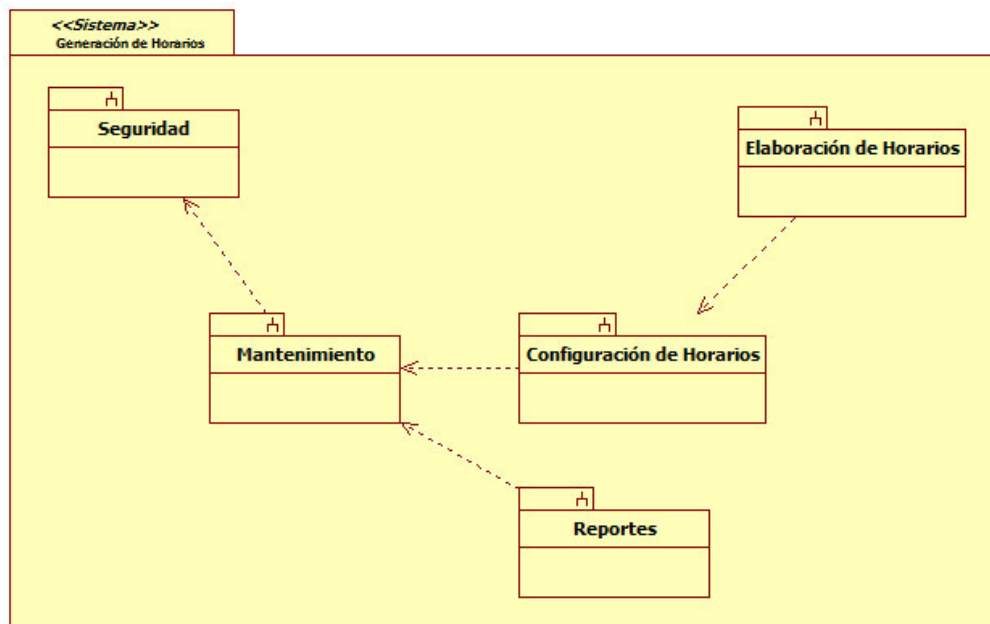


Figura 3 Diagrama de Vista Lógica del Sistema de Generación de Horarios [Fuente: Elaboración Propia].

5. Vista de Implementación

Se ha desarrollado el diagrama de Componentes del Sistema de Generación de Horarios en el siguiente diagrama:

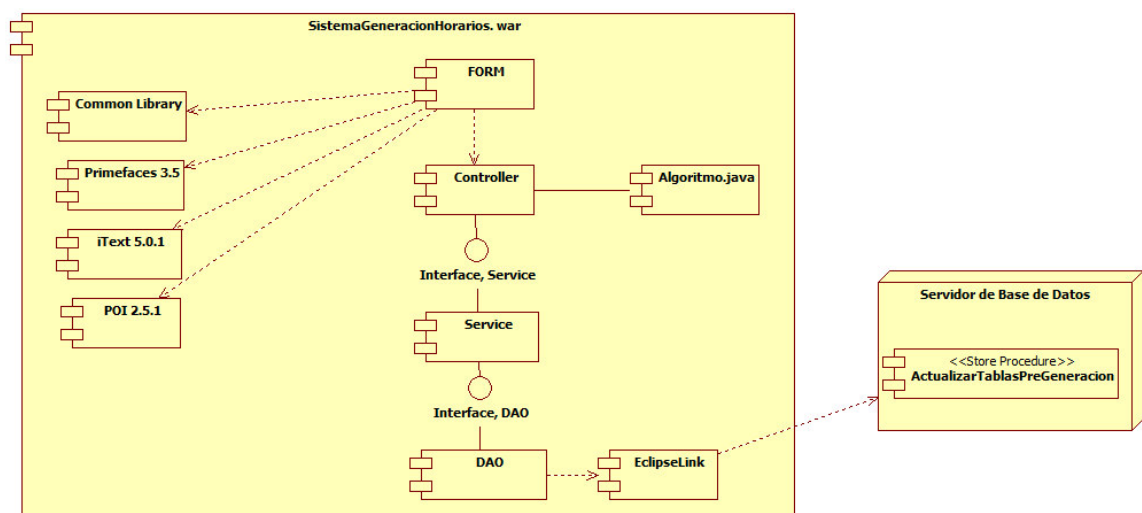


Figura 4 Diagrama de Componentes del Sistema de Generación de Horarios [Fuente: Elaboración Propia].

6. Vista de Despliegue

El software del Sistema de Generación de Horarios se desplegará en el servidor de la Facultad en una red interna de acceso solo para los Directores de Escuela, Jefes de Departamentos, Áreas de Logística y de Laboratorio y los Docentes, a continuación mostramos el diagrama de despliegue:

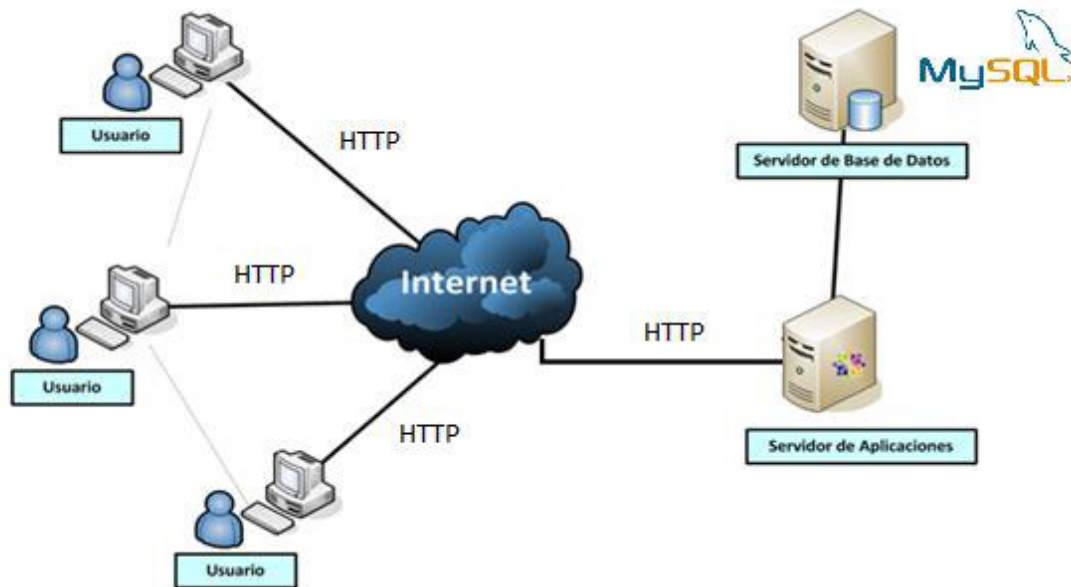


Figura 5 Diagrama de Despliegue del Sistema de Generación de Horarios [Fuente: Elaboración Propia].

7.1 Modelo Físico

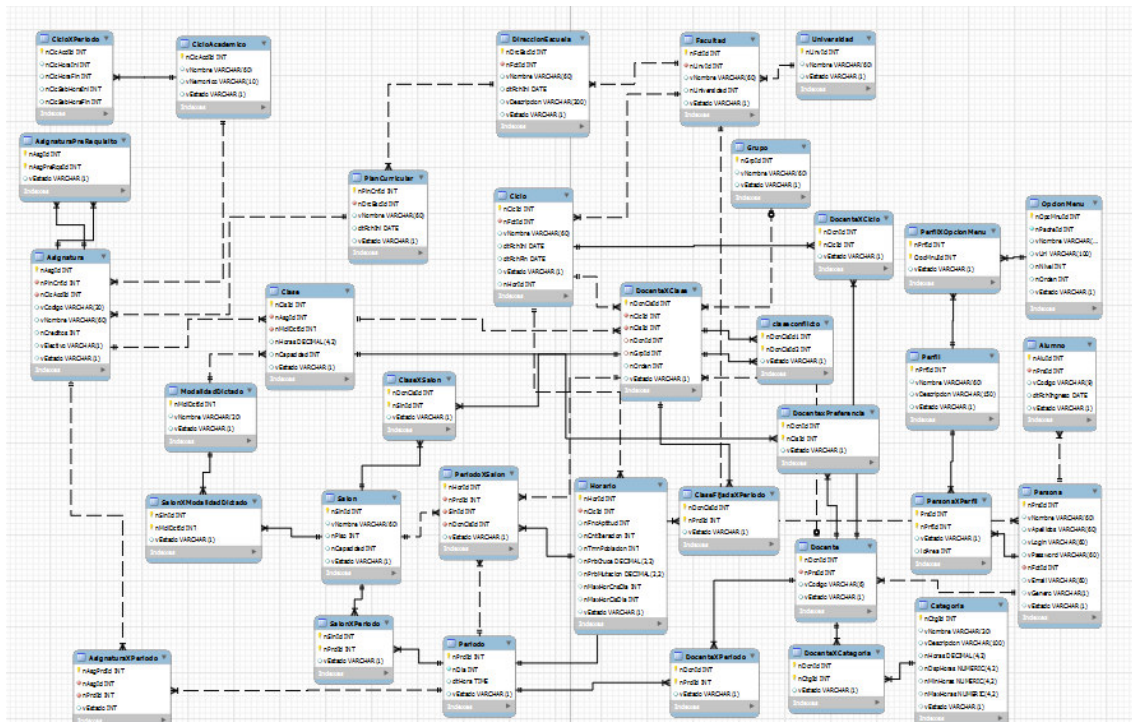


Figura 6 Modelo Físico del Sistema de Generación de Horarios [Fuente: Elaboración Propia].

ANEXO 4

MANUEL DE USUARIO

1. Introducción

El presente documento contiene la información necesaria para utilizar Sistema de Generación de Horarios para la Facultad de Sistemas de la Universidad Nacional Mayor de San Marcos

Se describirá todas las funcionalidades para facilitar la utilización del sistema a los usuarios describiendo todas las opciones. El sistema permite registrar la disponibilidad de los docentes, de las aulas, fijar dictados de clases de docentes externos a la facultad o docentes que deseen mantener su horario de dictado de clase, las direcciones de escuela, planes curriculares, asignaturas, clases, dictados de clases así como como configurar el periodo de dictados de clases de Asignaturas que pertenece a un determinado ciclo académico.

2. Objetivos y Alcance

2.1 Objetivo

El objetivo de este manual es proveer a los usuarios la información adecuada para la generación de horarios en los diferentes ciclos. Así como consultas simples de los mismos.

2.2 Alcance

El alcance de este manual es explicar las funcionalidades que tienen los módulos:

1. Mantenimiento

- Dirección de Escuela
- Plan de Estudios
- Asignaturas
- Clases
- Salones
- Grupos

2. Configuración de Horarios

- Ciclo
- Dictado de Clases

- Asignar Clases
- Disponibilidad Docente
- Configuración de Periodos de Dictado
- Configurar Parámetros.

3. Elaboración de Horarios

- Gestión de Horarios
- Visualizar Horarios

4. Seguridad

- Perfiles
- Usuario
- Cambiar Contraseña
- Cambiar Perfil

3. Descripción del Sistema

3.1 Descripción de acceso al sistema

En esta opción se describirá el ingreso del usuario al sistema, describiendo el flujo correcto y las validaciones.

Los pasos a seguir son los siguientes:

1. Para tener acceso al sistema se deberá ingresar el usuario y la clave que se le asignaron (Figura 1).

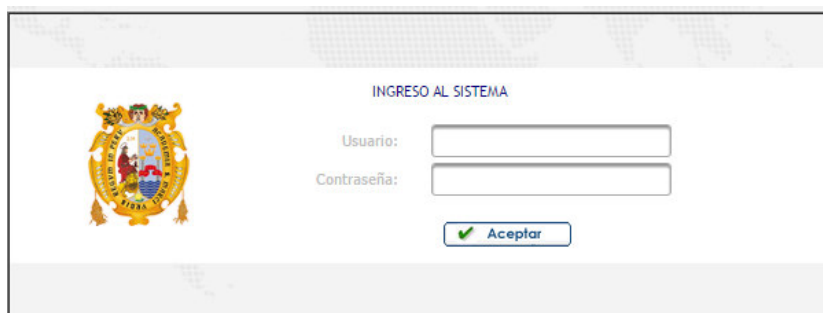


Figura 1 Pantalla de Acceso

Los campos correspondientes a la pantalla son los siguientes:

Usuario	Dato de entrada, que permite ingresar el código del usuario.
Clave	Dato de entrada, que permite ingresar la clave de acceso asignada.

2. Al ingresar una clave incorrecta se muestra el mensaje (Figura 2).



The screenshot shows a login interface titled "INGRESO AL SISTEMA". On the left is the coat of arms of the Republic of El Salvador. The login fields are labeled "Usuario:" and "Contraseña:". The "Usuario:" field contains the text "jpiedrai". The "Contraseña:" field is empty. To the right of the password field, a red error message states "La Contraseña es incorrecta". Below the fields is a green button with a checkmark icon and the text "Aceptar".

Figura 2 Mensaje de Clave Incorrecta

3. El sistema mostrará la pantalla del sistema de Generación de Horarios (Figura 3), en donde en la parte izquierda nos mostrará los 4 mencionados en 2.2, en la parte superior derecha nos mostrará información acerca del usuario logueado, su perfil, así como el ciclo vigente.



Figura 3 Sistema de Generación de Horarios

El Menú principal muestra los siguientes módulos:

- Mantenedores.
- Configuración de Horarios.
- Elaboración de Horarios.
- Seguridad.

3.2. Mantenimiento

Este módulo cuenta con las siguientes opciones:

- Dirección de Escuela
- Plan de Estudios
- Asignaturas
- Clases
- Salones
- Grupos

3.2.1. Dirección de Escuela

Este submenú (Figura 4) permite registrar y editar una Dirección de Escuela.



Figura 4 Dirección de Escuela

Los pasos a seguir son los siguientes:

1. Seleccione el menú “Mantenedores”, elija el submenú “Dirección de Escuela” y se mostrara la siguiente pantalla (Figura 5).



Figura 5 Mantenimiento de Dirección de Escuela

3.2.2. Plan de Estudios

Este submenú (Figura 6) permite registrar y editar un Plan de Estudios.



Figura 6 Plan de Estudios

Los pasos a seguir son los siguientes:

1. Seleccione el menú “Mantenedores”, elija el submenú “Plan de Estudios” y se mostrara la siguiente pantalla (Figura 7).

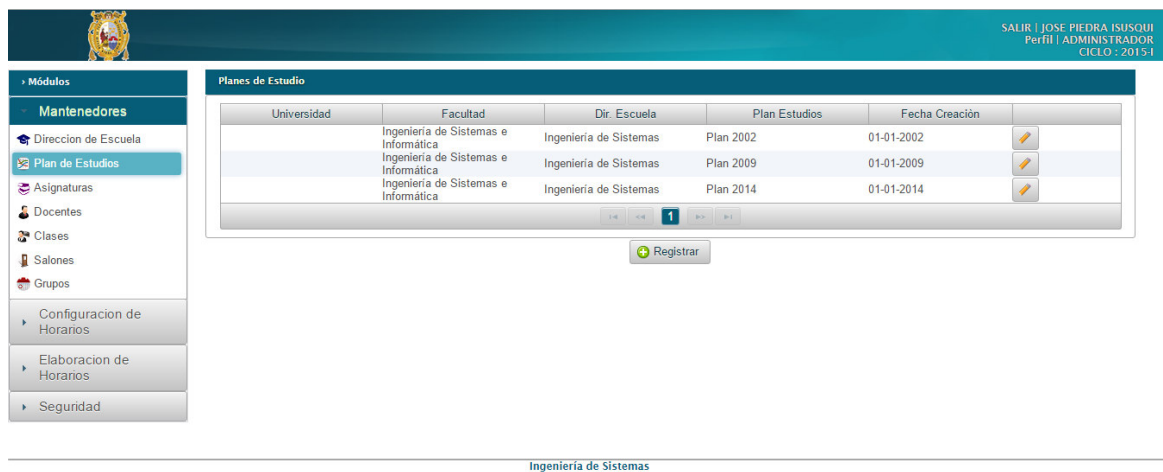


Figura 7 Mantenimiento de Plan de Estudios

2. Para registrar un Plan de Estudios seleccione la opción “Registrar”, le mostrará una ventana (Figura 8) en donde ingresará:

Nombre	Dato de entrada, que permite ingresar nombre del Plan de Estudios.
Dirección de Escuela	Dato de entrada, que permite seleccionar la Dirección de Escuela a la que pertenece el Plan de Estudio.
Fecha de Creación	Dato de entrada, que permite ingresar la fecha de creación del Plan de Estudio.

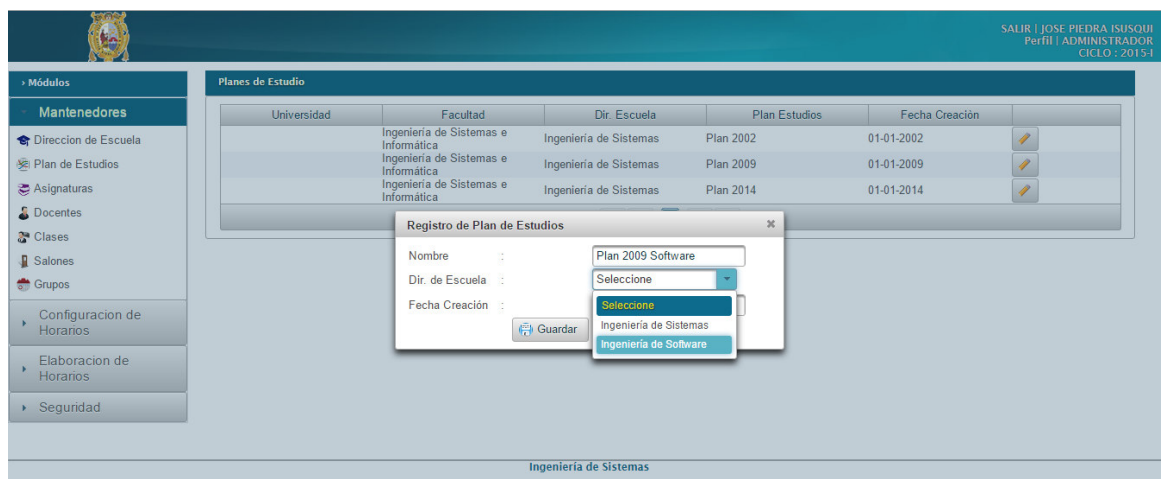


Figura 8 Registro de Plan de Estudios

3. Seleccionamos la opción Guardar y nos muestra un mensaje en donde nos indica que se creó satisfactoriamente como se muestra en la Figura 9.

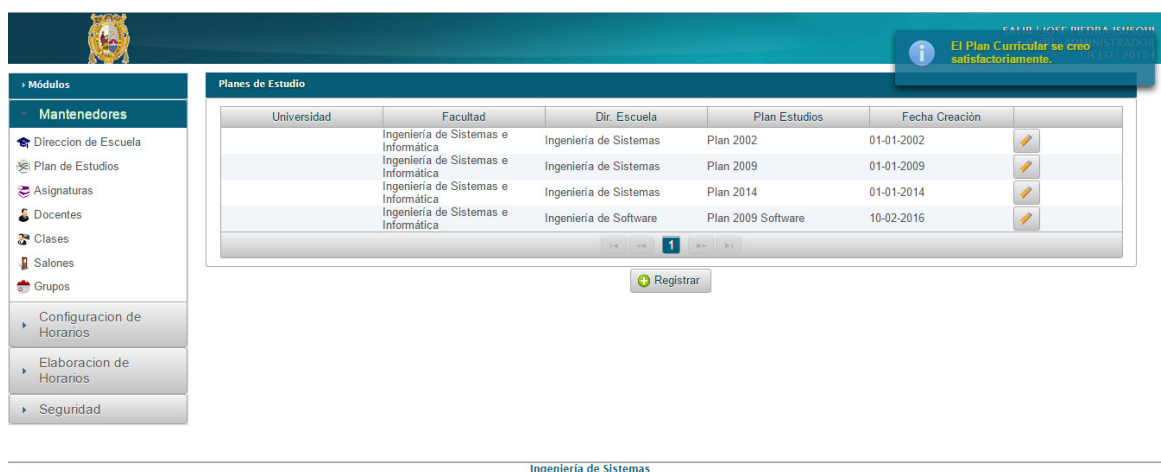


Figura 9 Registro satisfactorio de un Plan de Estudios

3.2.3. Asignaturas

Este submenú (Figura 10) permite registrar, editar y eliminar un Asignatura.



Figura 10 Asignaturas

Los pasos a seguir son los siguientes:

1. Seleccione el menú “Mantenedores”, elija el submenú “Asignaturas” y se mostrara la siguiente pantalla (Figura 11) en donde nos muestra una lista de Asignaturas ordenados de manera ascendente por Ciclo Académico pertenecientes a un Plan de Estudio con los campos Id, Código de Asignatura, Ciclo al que pertenece, Nombre de Asignatura, Nro. De Créditos.

The screenshot shows a web application interface. On the left is a sidebar menu with 'Módulos' and 'Mantenedores' expanded, showing 'Asignaturas' selected. The main area is titled 'Lista de Asignaturas - Plan de estudios : Plan 2009'. Below the title is a dropdown menu set to 'Selección Plan de Estudios: Plan 2009'. The table below lists 15 subjects with columns for Id, Código, Ciclo, Nombre, and Cred. Each row has edit and delete icons. At the bottom is a 'Registrar' button.

Id	Código	Ciclo	Nombre	Cred.
1	201001	CICLO I	Algoritmica I	4
2	201008	CICLO I	Computacion e Informatica	4
3	201003	CICLO I	Calculo I	4
4	201004	CICLO I	Matematica Basica I	4
5	201007	CICLO I	Teoria de Sistemas	3
6	201009	CICLO I	Taller de Tecnicas de Estudio	3
7	202001	CICLO II	Algoritmica II	6
8	202008	CICLO II	Estructura de Datos	4
9	202003	CICLO II	Calculo II	4
10	202007	CICLO II	Fisica General	4
11	202004	CICLO II	Matematica Basica II	4
12	202009	CICLO II	Economia	2
13	204001	CICLO III	Algoritmica III	4
14	204006	CICLO III	Diseño Grafico	3
15	204005	CICLO III	Estadistica I	4

Figura 11 Mantenimiento de Asignaturas

2. Para registrar una Asignatura seleccione la opción “Registrar”, le mostrará una ventana (Figura 12) en donde ingresará:

Código	Dato de entrada, que permite ingresar el código de Asignatura.
Nombre	Dato de entrada, que permite ingresar el nombre de la Asignatura.
Nro, Créditos	Dato de entrada, que permite ingresar el número de créditos de la Asignatura
Ciclo	Dato de entrada, que permite seleccionar el Ciclo Académico al que pertenece.

Salir | JOSE PIEDRA ISUSQUI
Perfil | ADMINISTRADOR
CICLO : 2013-I

Módulos

Mantenedores

- Dirección de Escuela
- Plan de Estudios
- Asignaturas
- Docentes
- Clases
- Salones
- Grupos
- Configuración de Horarios
- Elaboración de Horarios
- Seguridad

Lista de Asignaturas - Plan de estudios : Plan 2009

Seleccione Plan de Estudios: Plan 2009

Id	Código	Ciclo	Nombre	Cred.		
1	201001	CICLO I	Algoritmica I	4		
2	201008	CICLO I	Computacion e Informatica	4		
3	201003	CICLO I	Calculo I	4		
4	201004			4		
5	201007			3		
6	201009			3		
7	202001			6		
8	202008			4		
9	202003			4		
10	202007			4		
11	202004			4		
12	202009			2		
13	204001	CICLO III	Algoritmica III	4		
14	204006	CICLO III	Diseño Grafico	3		
15	204005	CICLO III	Estadistica I	4		

Registro de Nueva Asignatura

Código: 201009

Nombre: Asignatura de Prueba

Ncréditos: 4

Ciclo: Primer Ciclo

Guardar

Registrar

Figura 12 Registro de Asignaturas

- Seleccionamos la opción Guardar y nos muestra un mensaje en donde nos indica que se creó satisfactoriamente como se muestra en la Figura 13.

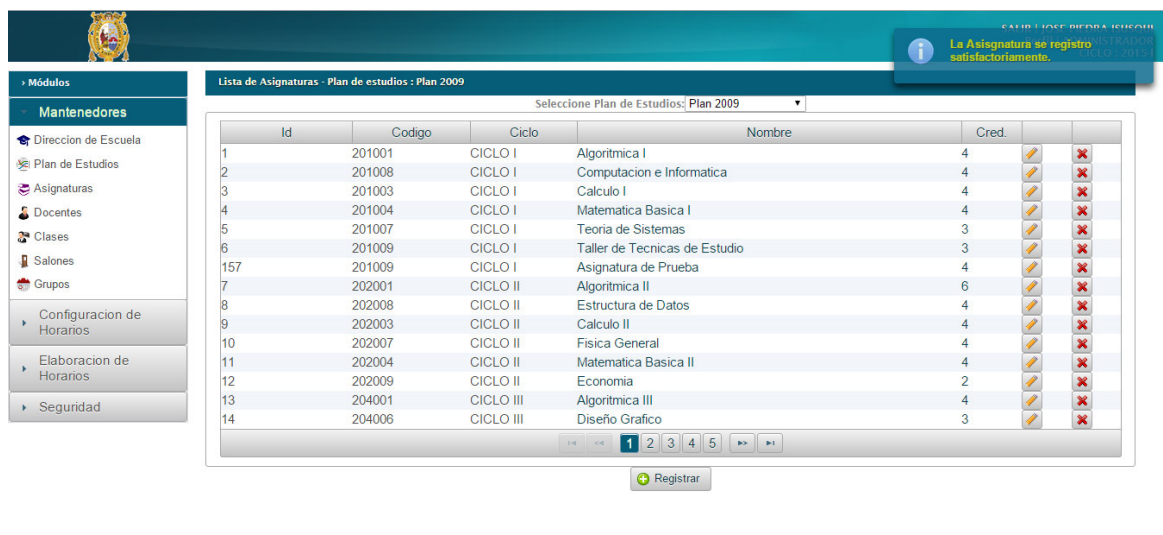


Figura 13 Registro exitoso de Asignaturas

3.2.4. Docentes

Este submenú (Figura 14) permite registrar, editar y eliminar un Docente.



Figura 14 Docentes

Los pasos a seguir son los siguientes:

1. Seleccione el menú “Mantenedores”, elija el submenú “Docentes” y se mostrara la siguiente pantalla (Figura 15) en donde nos muestra una lista de Docentes con los campos Código de Docente, Nombre, Apellido, Email.



Figura 15 Mantenimiento de Docentes

- Para registrar un Docente seleccione la opción “Registrar”, le mostrará una ventana (Figura 16) en donde ingresará:

Nombre	Dato de entrada, que permite ingresar el nombre del Docente.
Apellidos	Dato de entrada, que permite ingresar el apellido del Docente.
Email	Dato de entrada, que permite ingresar el email del Docente.
Categoría	Dato de entrada, que permite seleccionar la categoría del Docente.
Usuario	Dato de entrada, que permite ingresar el usuario del Docente.
Contraseña	Dato de entrada, que permite ingresar la contraseña del Docente.

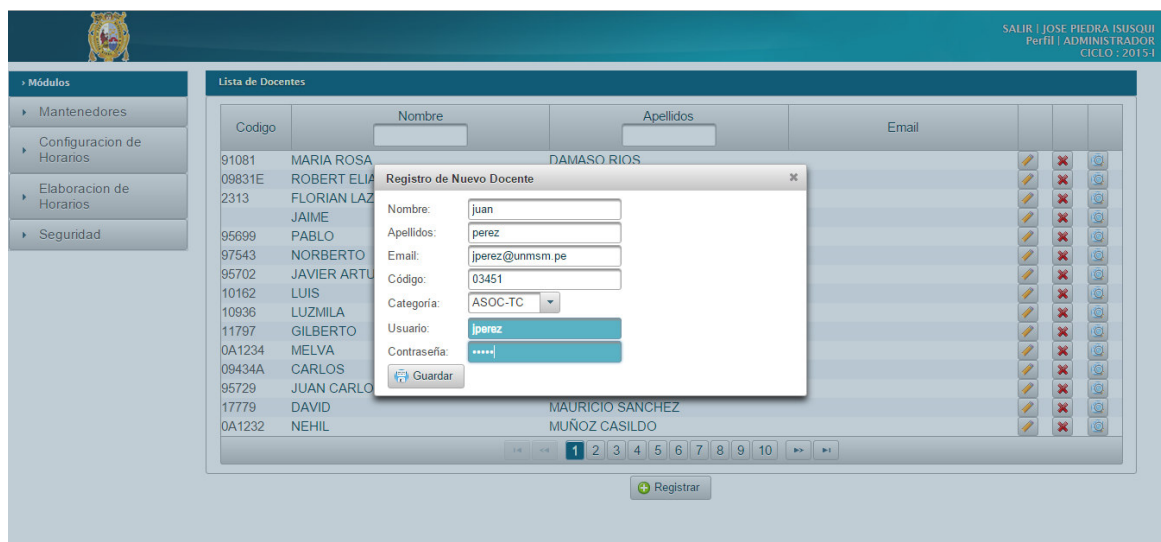


Figura 15 Registro de Docentes

3. Seleccionamos la opción Guardar y el docente de registra como se muestra en la Figura 16.



Figura 16 Registro satisfactorio de Docentes

4. Para registrar la disponibilidad de docente seleccionamos la opción disponibilidad (icono de lupa) en donde nos mostrará una ventana como se muestra en la Figura 17.

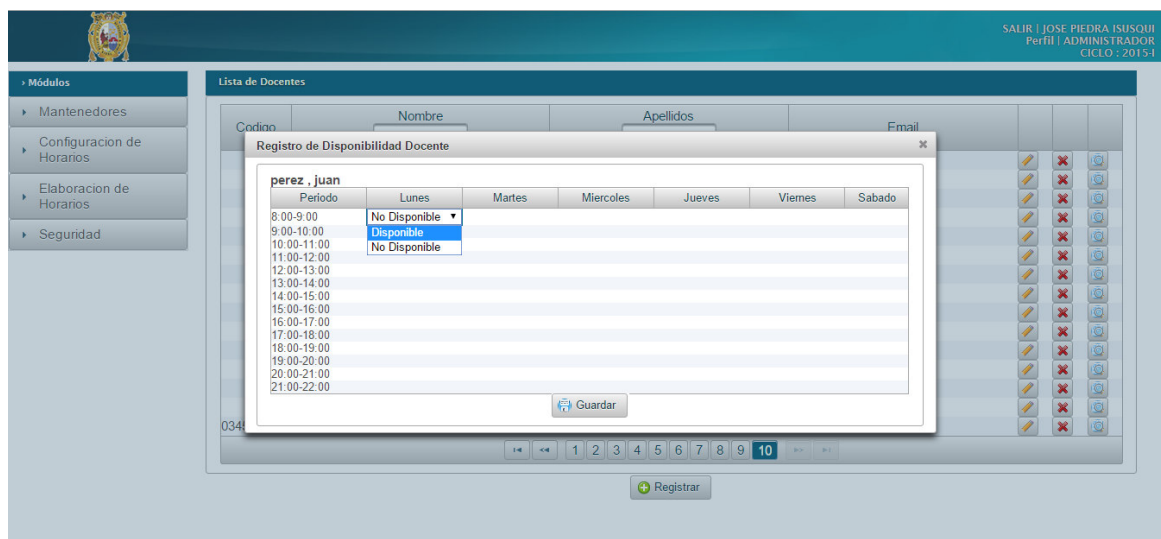


Figura 16 Registro de disponibilidad de Docentes

5. Seleccionamos la opción Guardar y nos muestra un mensaje en donde nos indica que se actualizó con éxito la disponibilidad del docente como se muestra en la Figura 17.



Figura 17 Registro satisfactorio de disponibilidad de Docentes

3.2.5. Clases


Este submenú (Figura 18) permite registrar, editar y eliminar una Clase.



Figura 18 Clases

Los pasos a seguir son los siguientes:

1. Seleccione el menú “Mantenedores”, elija el submenú “Clases” y se mostrara la siguiente pantalla (Figura 19) en donde nos muestra una lista de Clases ordenados de manera ascendente por Ciclo Académico pertenecientes a un Plan de Estudio con los campos Ciclo Académico, Nombre de la Asignatura a las que pertenece la clase, Tipo de Dictado (Teórico, Práctico, Laboratorio), N de horas, capacidad (debe ser menor o igual a la capacidad de un salón para que pueda ser asignado este), Estado.



SALIR | JOSE PIEDRA ISUSQUI
Perfil | ADMINISTRADOR
CICLO : 2015-I

Módulos

Mantenedores

Escuela

Plan de Estudios

Asignaturas

Docentes

Clases

Salones

Grupos

Configuración de Horarios

Ciclo

Dictados de Clases

Asignar Clases

Disponibilidad Docente

Conf. Periodo de Dictado

Configurar Parametros

Clases de Preferencia

Lista de Clases - Plan de estudios : Plan 2009

Seleccione Plan de Estudios: Plan 2009

Ciclo	Asignatura	Tipo de Dictado	Nº de horas	Capacidad	Estado		
CICLO I	Algoritmica I	Teórico	2.00 h.	40	1		
CICLO I	Algoritmica I	Práctico	2.00 h.	40	1		
CICLO I	Algoritmica I	Laboratorio	2.00 h.	20	1		
CICLO I	Computacion e Informatica	Teórico	3.00 h.	40	1		
CICLO I	Computacion e Informatica	Laboratorio	2.00 h.	20	1		
CICLO I	Calculo I	Teórico	3.00 h.	40	1		
CICLO I	Calculo I	Práctico	2.00 h.	40	1		
CICLO I	Matematica Basica I	Teórico	3.00 h.	40	1		
CICLO I	Matematica Basica I	Práctico	2.00 h.	40	1		
CICLO I	Teoria de Sistemas	Teórico	3.00 h.	30	1		
CICLO I	Taller de Tecnicas de Estudio	Teórico	3.00 h.	20	1		
CICLO II	Algoritmica II	Teórico	2.00 h.	40	1		
CICLO II	Algoritmica II	Práctico	2.00 h.	40	1		
CICLO II	Algoritmica II	Laboratorio	2.00 h.	20	1		
CICLO II	Estructura de Datos	Teórico	3.00 h.	30	1		
CICLO II	Estructura de Datos	Práctico	2.00 h.	30	1		
CICLO II	Calculo II	Teórico	3.00 h.	40	1		
CICLO II	Calculo II	Práctico	2.00 h.	40	1		
CICLO II	Fisica General	Teórico	2.00 h.	40	1		
CICLO II	Fisica General	Práctico	2.00 h.	40	1		

localhost:8081/SistemasHorariosClienteWeb/clase#

Figura 19 Mantenimiento de Clases

- Para registrar una clase seleccione la opción “Registrar”, le mostrará una ventana (Figura 20) en donde ingresará:

Asignatura	Dato de entrada, que permite ingresar el nombre del Docente.
Tipo de Dictado	Dato de entrada, que permite ingresar el apellido del Docente.
Cantidad de Horas	Dato de entrada, que permite ingresar el email del Docente.
Capacidad	Dato de entrada, que permite seleccionar la categoría del Docente.

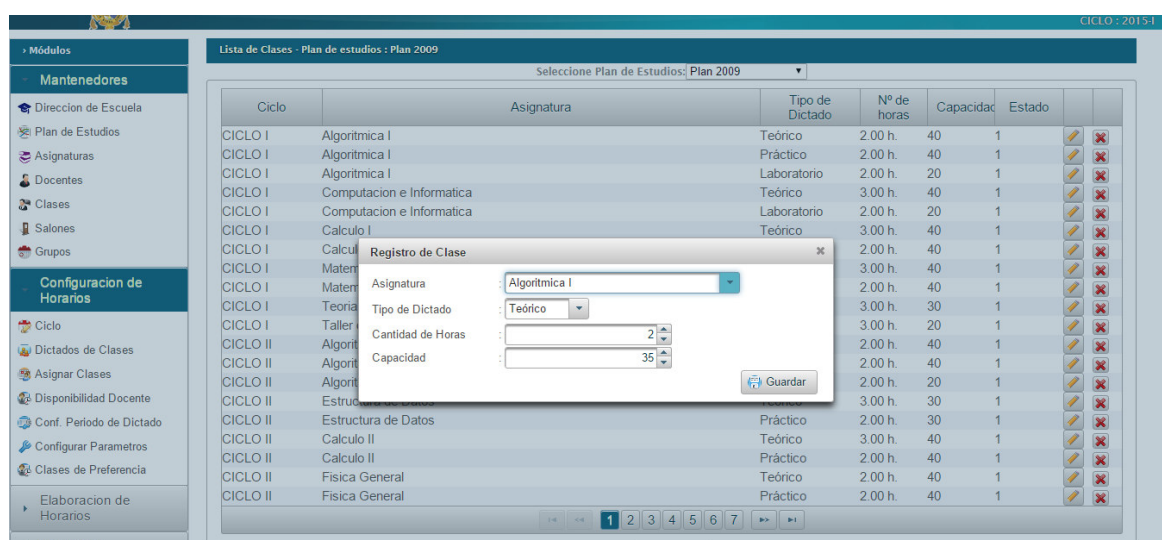


Figura 20 Registro de Clases

- Seleccionamos la opción Guardar y nos muestra un mensaje en donde nos indica que la clase se registró satisfactoriamente como se muestra en la Figura 21.

Dirección de Escuela	Ciclo	Asignatura	Tipo de Dictado	Nº de horas	Capacidad	Estado		
Plan de Estudios	CICLO I	Algoritmica I	Teórico	2.00 h	40	1		
Asignaturas	CICLO I	Algoritmica I	Teórico	2 h	40	1		
Docentes	CICLO I	Algoritmica I	Práctico	2.00 h	40	1		
Clases	CICLO I	Algoritmica I	Laboratorio	2.00 h	20	1		
Salones	CICLO I	Computacion e Informatica	Teórico	3.00 h	40	1		
Grupos	CICLO I	Computacion e Informatica	Laboratorio	2.00 h	20	1		
Configuracion de Horarios	CICLO I	Calculo I	Teórico	3.00 h	40	1		
Ciclo	CICLO I	Calculo I	Práctico	2.00 h	40	1		
Dictados de Clases	CICLO I	Matematica Basica I	Teórico	3.00 h	40	1		
Asignar Clases	CICLO I	Matematica Basica I	Práctico	2.00 h	40	1		
Disponibilidad Docente	CICLO I	Teoria de Sistemas	Teórico	3.00 h	30	1		
Conf. Periodo de Dictado	CICLO I	Taller de Tecnicas de Estudio	Teórico	3.00 h	20	1		
Configurar Parametros	CICLO II	Algoritmica II	Teórico	2.00 h	40	1		
Clases de Preferencia	CICLO II	Algoritmica II	Práctico	2.00 h	40	1		
Elaboracion de Horarios	CICLO II	Algoritmica II	Laboratorio	2.00 h	20	1		
Seguridad	CICLO II	Estructura de Datos	Teórico	3.00 h	30	1		
	CICLO II	Estructura de Datos	Práctico	2.00 h	30	1		
	CICLO II	Calculo II	Teórico	3.00 h	40	1		
	CICLO II	Calculo II	Práctico	2.00 h	40	1		
	CICLO II	Fisica General	Teórico	2.00 h	40	1		

Figura 21 Registro satisfactorio de Clases

3.2.6. Salones

Este submenú (Figura 22) permite registrar, editar y eliminar un salón.



Figura 22 Salones

Los pasos a seguir son los siguientes:

1. Seleccione el menú “Mantenedores”, elija el submenú “Salones” y se mostrara la siguiente pantalla (Figura 23) en donde nos muestra una lista de Salones con los campos Nombre, N° piso, capacidad, Estado.



Figura 23 Mantenimiento de Salones

- Para registrar un salón seleccione la opción “Registrar”, le mostrará una ventana (Figura 24) en donde ingresará:

Nombre	Dato de entrada, que permite ingresar el nombre del Docente.
N° piso	Dato de entrada, que permite ingresar el número de piso.
Capacidad	Dato de entrada, que permite ingresar la capacidad del salón
Tipo de Dictado	Dato de entrada, que permite seleccionar el Tipo de Dictado (Teórico, Práctico, Laboratorio)
Estado	Dato de entrada, que permite ingresar el Estado (Disponible o No Disponible) para ser considerado para en el algoritmo

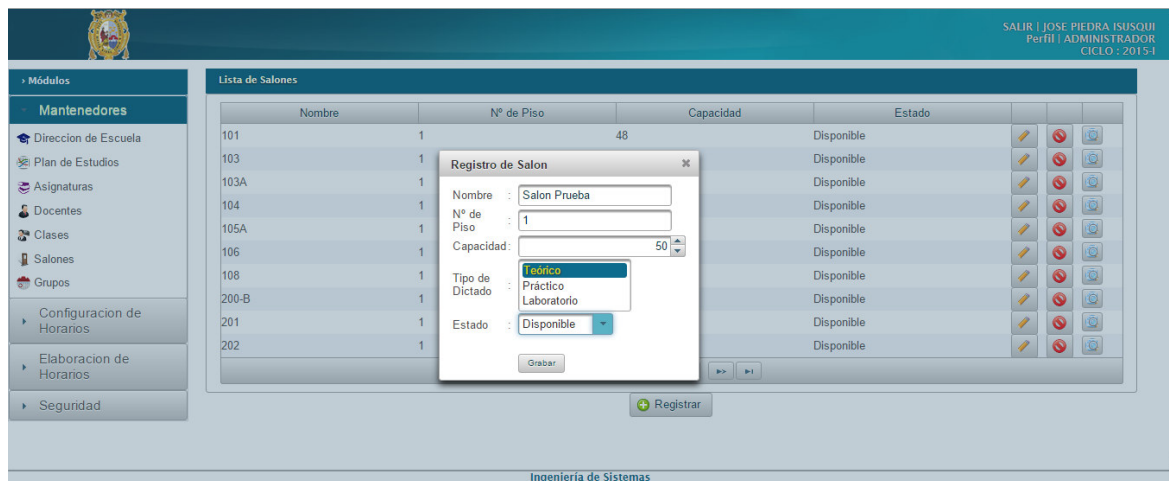


Figura 24 Registro de Salones

3. Seleccionamos la opción Grabar y se registra como se muestra en la Figura 25.

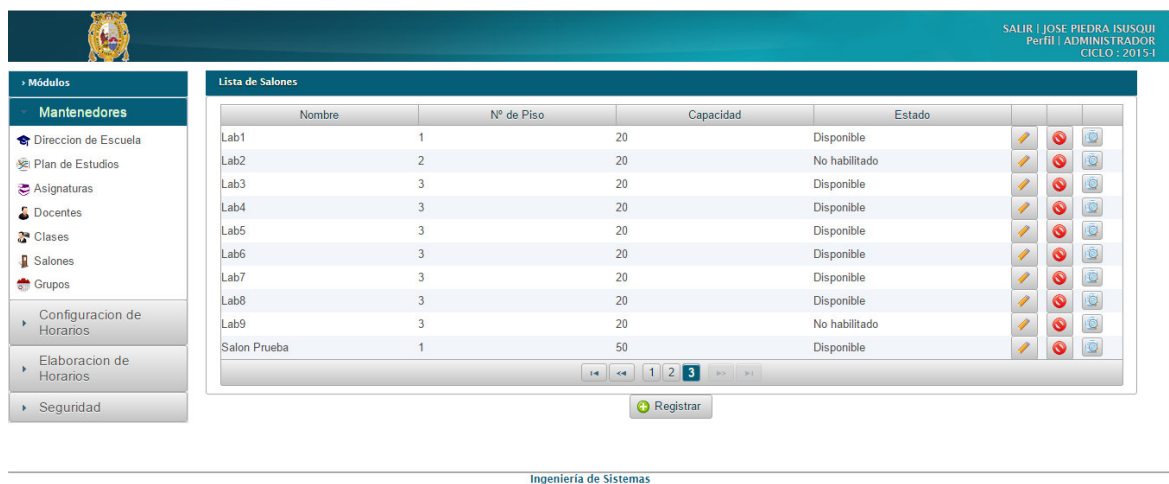


Figura 25 Registro satisfactorio de Salones

4. Para registrar la disponibilidad del salón seleccionamos la opción disponibilidad (icono de lupa) en donde nos mostrará una ventana como se muestra en la Figura 26.

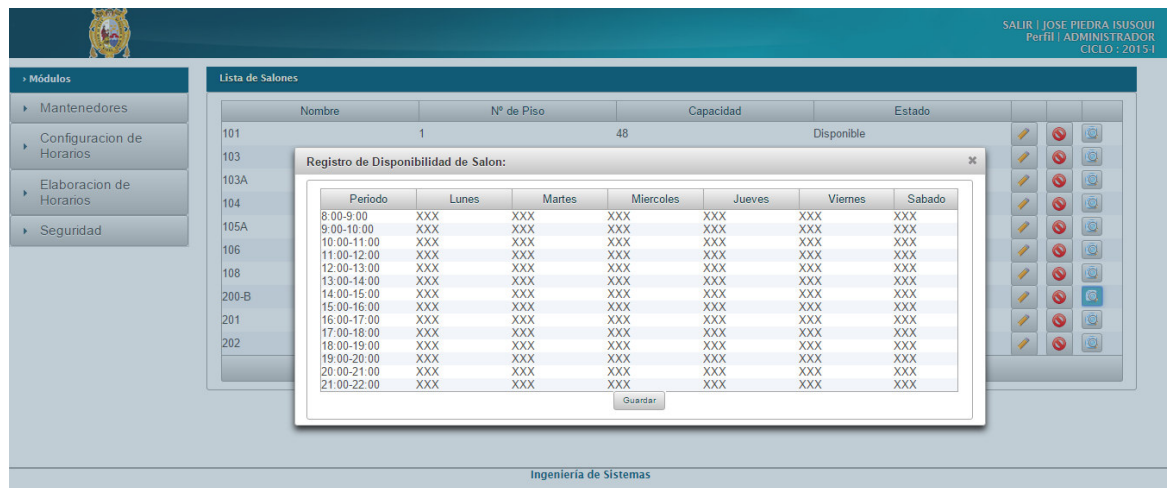


Figura 26 Registro de disponibilidad de Salones

5. Seleccionamos la opción Guardar para actualizar la disponibilidad del Salón.

3.2.7. Grupos

Este submenú (Figura 27) permite registrar, editar y eliminar un grupo.



Figura 27 Grupos

Los pasos a seguir son los siguientes:

1. Seleccione el menú “Mantenedores”, elija el submenú “Grupos” y se mostrara la siguiente pantalla (Figura 28) en donde nos muestra una lista de Grupos con los campos Nombre, Estado.

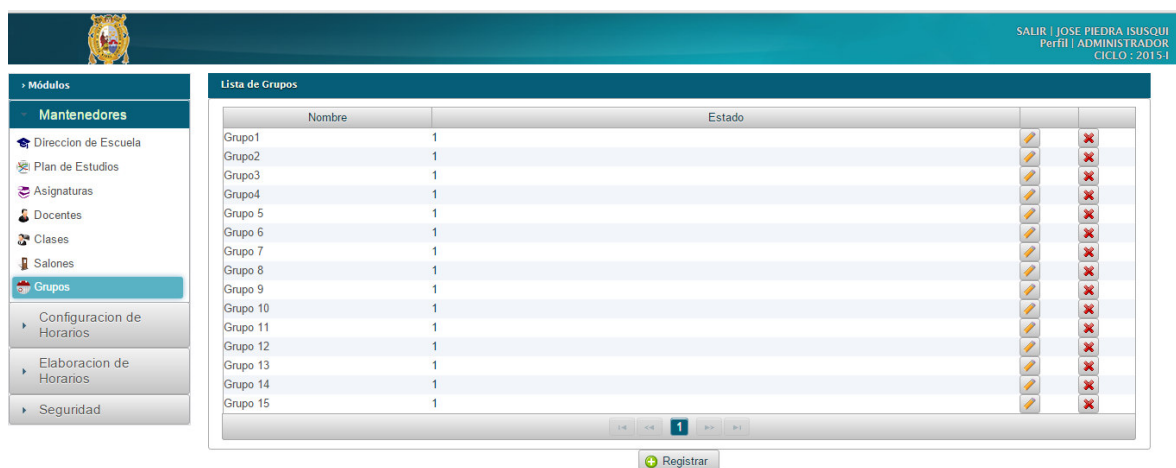


Figura 28 Mantenimiento de Grupos

3.3. Configuración de Horarios

Este módulo cuenta con las siguientes opciones:

- Ciclo
- Dictado de Clases
- Asignar Clases
- Disponibilidad Docente
- Configuración de Periodos de Dictado

3.3.1. Ciclo

Este submenú (Figura 29) permite registrar y editar un Ciclo.

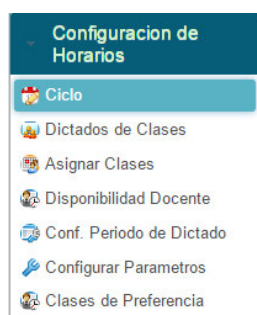


Figura 29 Ciclo

Los pasos a seguir son los siguientes:

1. Seleccione el menú “Configuración de Horarios”, elija el submenú “Ciclo” y se mostrara la siguiente pantalla (Figura 30).

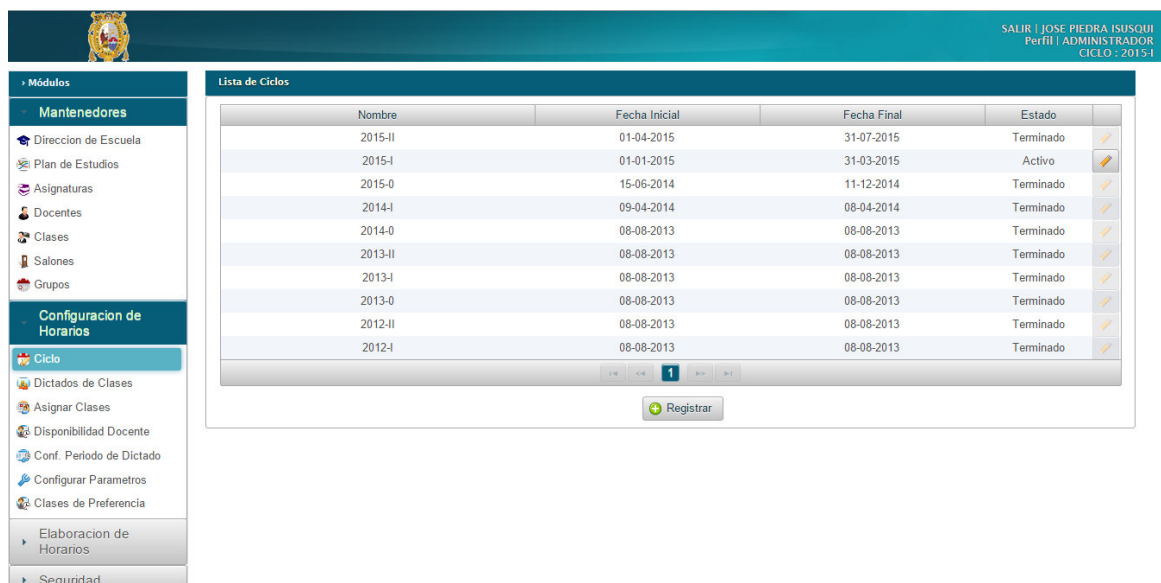


Figura 30 Mantenimiento de Ciclo

2. Para registrar un ciclo seleccione la opción “Registrar”, le mostrará una ventana (Figura 31) en donde ingresará:

Nombre	Dato de entrada, que permite ingresar el nombre del Docente.
Fecha Inicial	Dato de entrada, que permite ingresar la fecha en que inicia el ciclo.
Fecha Final	Dato de entrada, que permite ingresar la fecha en que termina el ciclo.
Estado	Dato de entrada, que permite seleccionar el Estado (Activo o Por Iniciar).

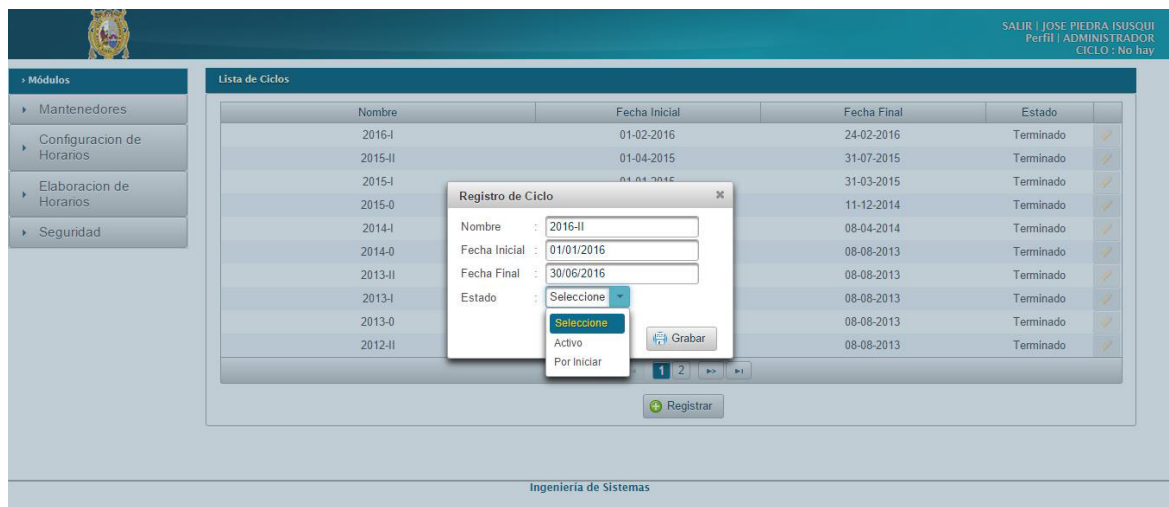


Figura 31 Registro de Ciclo

3. Seleccionamos la opción Grabar, se muestra un mensaje de éxito y se registra como se muestra en la Figura 32.

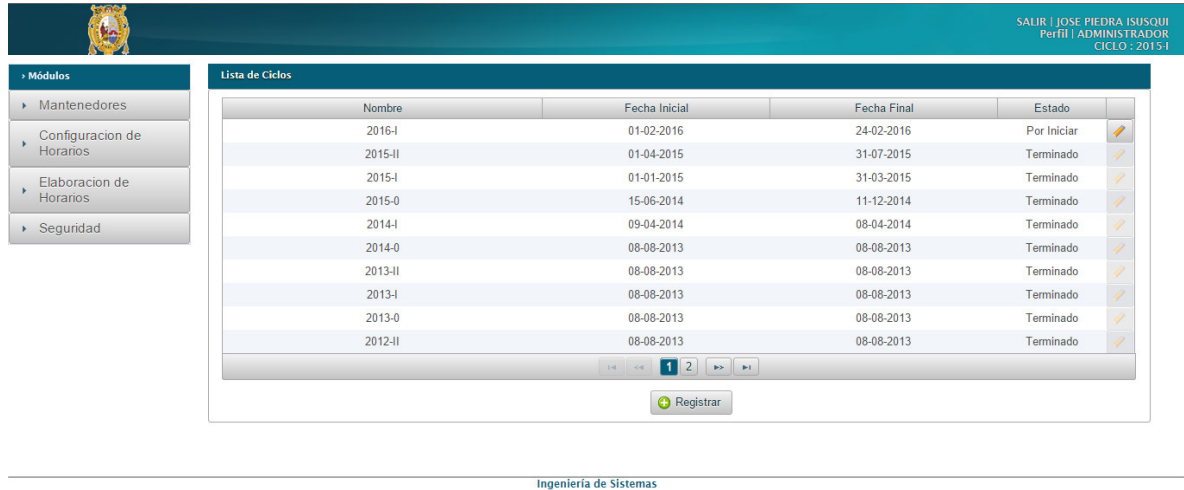


Figura 32 Registro satisfactorio de Ciclo